PanaXSeries

The One to Watch for Constant Innovation-Making the Future Come Alive

MICROCOMPUTER MN1030

MN103001G/F01K
LSI User's Manual

Pub.No.23101-050E

Panasonic

Request for your special attention and precautions in using the technical informaition and semiconductors described in this book

(1) An export permit needs to be obtained from the competent authorities of the Japanese Government if any of the products or technologies described in this book and controlled under the "Foreign Exchange and Foreign Trade Law" is to be exported or taken out of Japan.

(2) The contents of this book are subject to change without notice in matters of improved function.When finalizing your design, therefore, ask for the most up-to-date version in advance in order to check for any changes.

(3) We are not liable for any damage arising out of the use of the contents of this book, or for any infringement of patents or any other rights owned by a third party.

(4) No part of this book may be reprinted or reproduced by any means without written permission from our company.

(5) This book deals with standard specification. Ask for the latest individual Product Standards or Specifications in advance for more detailsd infomation required for your design, purchasing and applications.

If you have any inquiries or questions about this book or our semiconductors, please contact one of our sales offices listed at the back of this book.

Table of Contents/List of Figures and Tables

Table of Contents

1. General Specifications

2. CPU

3. Extension Instruction Specifications

4. Memory Modes

5. Operating Mode

6. Clock Generator

7. Internal Memory

8. Bus Controller (BC)

11. 16-bit Timers

12. Watchdog Timer

13. Serial Interface

14. A/D Converter

15. I/O Ports

List of Figures and Tables

List of Figures

1. General Specifications

2. CPU

3. Extension Instruction Specifications

4. Memory Modes

5. Operating Mode

6. Clock Generator

7. Internal Memory

8. Bus Controller (BC)

9. Interrupt Controller

10. 8-bit Timers

11. 16-bit Timers

12. Watchdog Timer

13. Serial Interface

14. A/D Converter

15. I/O Ports

16. Internal Flash Memory

17. Ordering Mask ROM

Appendix

List of Tables

1. General Specifications

2. CPU

3. Extension Instruction Specifications

4. Memory Modes

5. Operating Mode

6. Clock Generator

8. Bus Controller (BC)

9. Interrupt Controller

10. 8-bit Timers

11. 16-bit Timers

12. Watchdog Timer

13. Serial Interface

14. A/D Converter

15. I/O Ports

16. Internal Flash Memory

1. General Specifications

1.1 Overview

The MN1030 Series is a 32-bit microcontroller that maintains the software assets of Matsushita Electronics' 16-bit MN102 Series of microcontrollers by offering ease of use and excellent cost-performance with a simple, high-performance architecture.

Built around a compact 32-bit CPU core with a basic instruction word length of one byte, the MN103001G (mask ROM version) includes ROM, RAM, a bus control circuit, interrupt control circuit, timers, a serial interface, A/D converter, and input/output ports in a 100-pin QFP. The MN1030F01K (flash memory version) is equipped with flash memory instead of mask ROM, has the same on-chip peripheral functions as the MN103001G, and has the same package and pin specifications. This microcontroller is ideal for multimedia devices, which must be able to process large volumes of data (for audio, stills, video, etc.), as well as for real-time control equipment that requires fast and precise control. When supplied with power supply voltage of 3.3 V, the MN103001G operates at 60 MHz and achieves performance of 60 MIPS.

1.2 Features

Low voltage, high-speed processing, low power consumption

- ■ Minimum instruction execution time:
  16.7 ns (during 3.3 V internal 60 MHz operation *MN103001G)
  25 ns (during 3.3 V internal 40 MHz operation *MN1030F01K)
- ■ Power consumption (TYP.):
  300 mW (during 3.3 V internal 60 MHz operation *MN103001G)
  270 mW (during 3.3 V internal 40 MHz operation *MN1030F01K)

Compact and high-performance CPU core

- ■ Simple and highly efficient instruction set
  (Number of basic instructions: 46; number of extension instructions: 24; number of addressing modes: 6)
- ■ Excellent coding efficiency with instructions that have a basic word length of one byte
- ■ Load/store architecture with 5-stage pipeline organization provides fast instruction execution
- ■ High-speed branch processing
- ■ Supports linear address space of up to 4 GB
  (External extension 8 Mbytes x 4 = 32 Mbytes)

Extension operation functions

- ■ Multiply-and-accumulate operation instructions (32 bits x 32 bits + 64 bits = 64 bits; throughput: 2 clocks)
- ■ Saturation operation instructions
- ■ Bit search instructions
- ■ Swap instructions

Large on-chip memory

- ■ 128 Kbytes of on-chip ROM/256 Kbytes of flash memory
- ■ 8 Kbytes of on-chip RAM (for data storage)

Flexible clock control

- ■ Self-excited/externally excited oscillation
  - Maximum 60 MHz internal operation when a 15-MHz oscillator is connected or a 15-MHz clock is input (in the case of the MN103001G)
  - Can switch between using PLL (programmable: multiply by four, multiply by two, multiply by one)/not using PLL (divide by two)
- ■ Low power consumption mode
  - HALT, STOP, SLEEP mode

High-speed/high-performance bus interface

- ■ Can select either separate address/data buses or multiplex address/data bus
  - • Address: 24 bits/Data: 8/16 bits
- ■ External memory space can be partitioned into four blocks
  - • Chip select signal output for each block
  - • Blocks 2 to 3 can be switched between fixed wait insertion or handshaking
  - • Blocks 0 to 3 can be switched between synchronous mode and asynchronous mode
  - • Blocks 1 and 2 can be used as DRAM space
- ■ DRAM control circuit on chip
  - • Address multiplexing function
  - • Programmable RAS/CAS timing setting
  - • Refresh control
    - CAS-before-RAS refresh support
    - Programmable refresh interval
  - • High-speed page mode support
- ■ One store buffer on chip
  - • Avoids time penalty when performing a store operation in an internal peripheral or an external device

Input/output interface

- ■ Supports 3.3 V, CMOS-level input/output interface

Wide variety of internal peripheral functions

- ■ Interrupts
  - • 38 sources
    - External interrupts: 9 sources (IRQn (n=7 to 0) x 8, and $\overline{\text{NMIRQ}}$ x 1)
    - Internal interrupts: 29 sources (timers: 18; Serial I/F: 8; WDT: 1; A/D: 1; system error: 1)
- ■ Timers
  - • Twelve 8-bit timers (all are down-counters)
    - Format: Reload timer
    - Cascaded connection possible (permits use as 16- to 32-bit timers)
    - Timer output possible (duty ratio; 1:1,12 outputs)
    - PWM output possible (8 outputs)
    - Internal clock source or external clock source can be selected
    - Serial interface clock generation
    - A/D converter start timing generation
  - • One 16-bit timer (up-counter)
    - Internal clock source or external clock source can be selected
    - Input capture function (rising edge, falling edge, or both edges can be selected)
    - PWM generation functions
    - 2 compare and capture registers
  - • Three 16-bit timers (down-counter)
    - Format: reload timer
    - Internal clock source or external clock source can be selected
  - • One watchdog timer
- ■ Serial interface
  - • UART/synchronous system/I2C (multipurpose) x 1 channel
  - • UART-serial interface x 1 channel (maximum bit rate: 230.4 kbit/s)
  - • Synchronous x 2 channels
- ■ A/D converter
  - • 10 bits: 4 inputs
    - Automatic scanning possible (0 to 3 channels can be set)

- ■ Input ports:
  - 4 (all multipurpose)
- ■ Output ports:
  - 15 (all multipurpose)
- ■ Input/output ports:
  - 53 (all multipurpose)

Flash microcontroller specifications

- ■ Performance identical to that of a mask ROM product guaranteed
- ■ Overwriting while on board possible through serial communications
- ■ Batch/block erase possible

  Block units 8 KB (multiple blocks can be selected simultaneously)

Package

- LQFP100-P-1414

1.3 Block Diagram

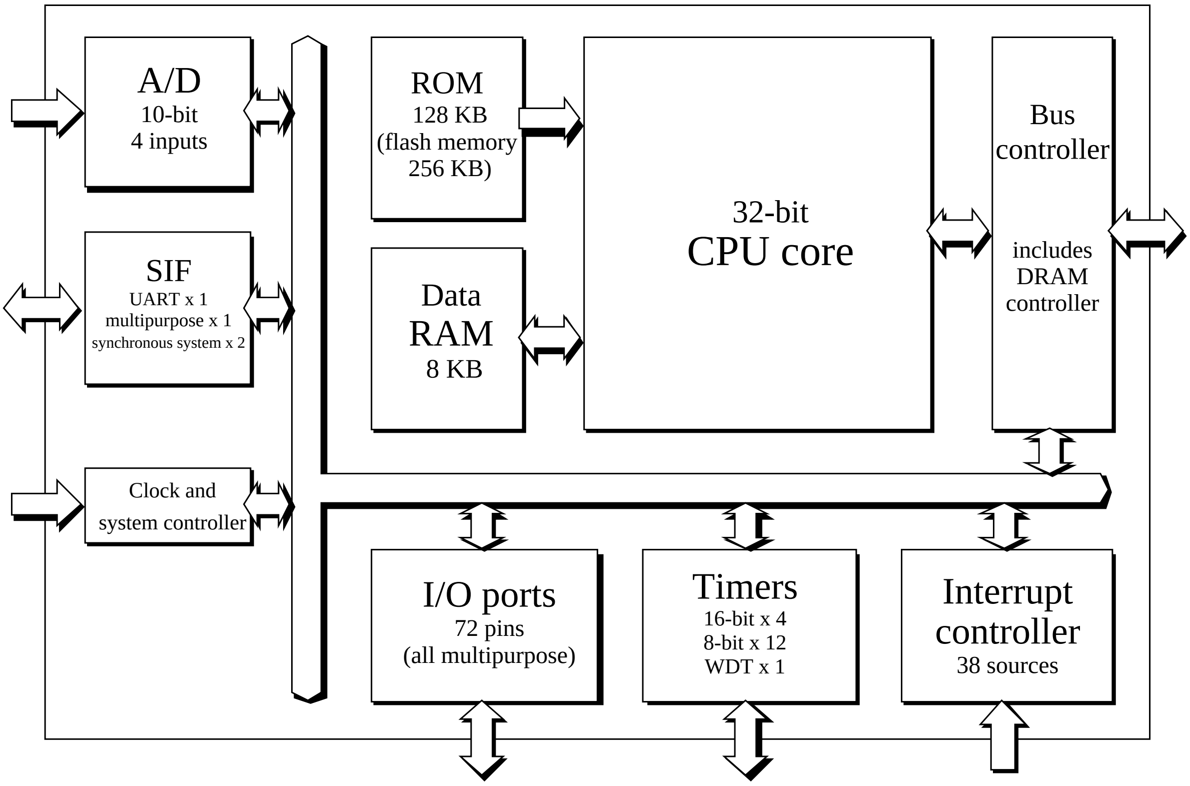


Fig. 1-3-1 MN103001G Block Diagram

* The MN1030F01K (flash version) is equipped with 256 KB of flash memory instead of 128 KB of ROM.

1.4 Pin Description

1.4.1 Pin Assignments

The pin assignments are shown in Fig. 1-4-1 and Table 1-4-1.

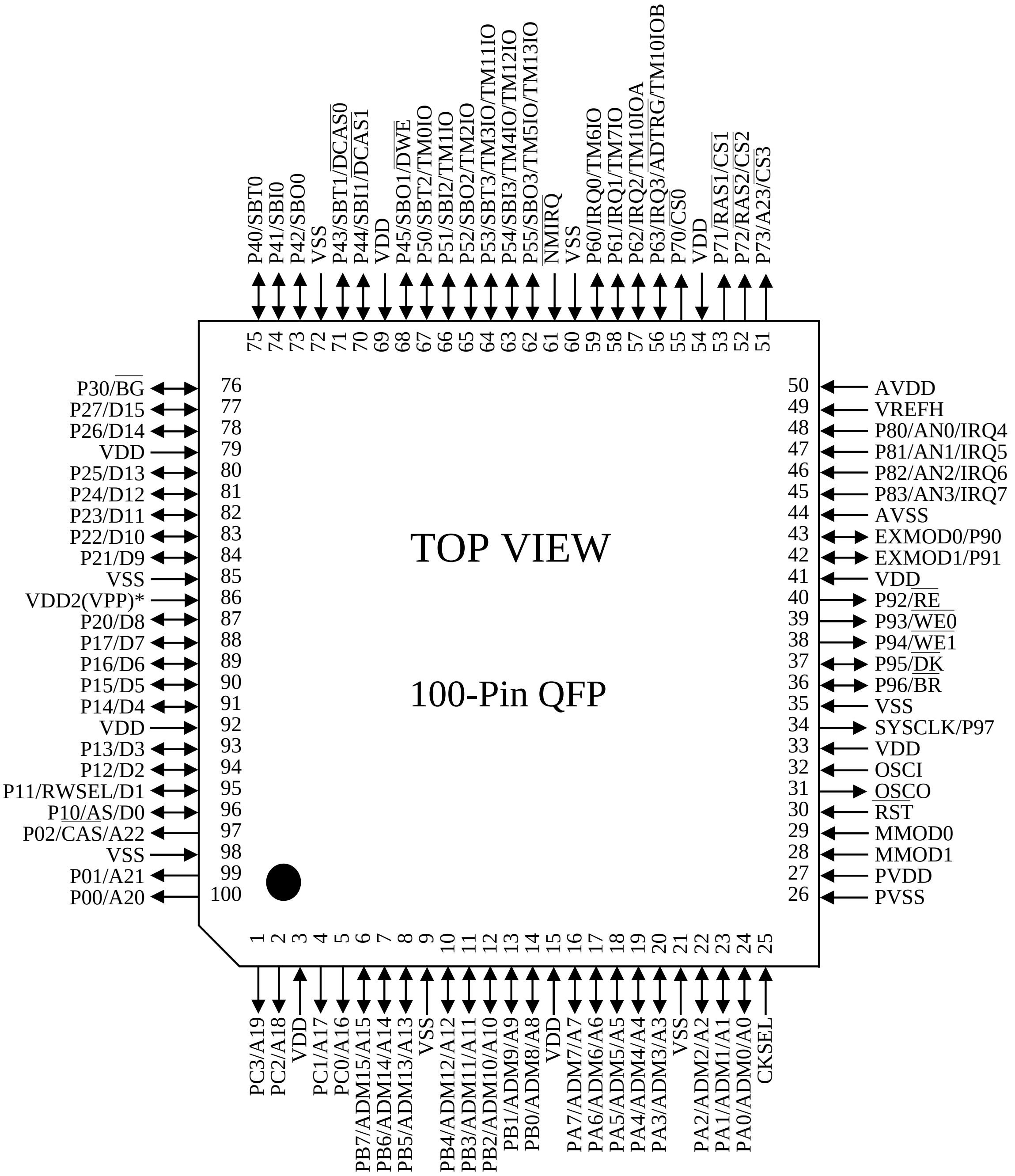


Fig. 1-4-1 Pin Assignments Diagram

* "VDD2" in the case of the MN103001G, "VPP" in the case of the MN1030F01K.

Table 1-4-1 Pin Assignments

Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name
1	PC3/A19	26	PVSS	51	P73/A23/$\overline{\text{CS3}}$	76	P30/$\overline{\text{BG}}$
2	PC2/A18	27	PVDD	52	P72/$\overline{\text{RAS2}}$/$\overline{\text{CS2}}$	77	P27/D15
3	VDD	28	MMOD1	53	P71/$\overline{\text{RAS1}}$/$\overline{\text{CS1}}$	78	P26/D14
4	PC1/A17	29	MMOD0	54	VDD	79	VDD
5	PC0/A16	30	$\overline{\text{RST}}$	55	P70/$\overline{\text{CS0}}$	80	P25/D13
6	PB7/ADM15/A15	31	OSCO	56	P63/IRQ3/$\overline{\text{ADTRG}}$/TM10IOB	81	P24/D12
7	PB6/ADM14/A14	32	OSCI	57	P62/IRQ2/TM10IOA	82	P23/D11
8	PB5/ADM13/A13	33	VDD	58	P61/IRQ1/TM7IO	83	P22/D10
9	VSS	34	SYSCLK/P97	59	P60/IRQ0/TM6IO	84	P21/D9
10	PB4/ADM12/A12	35	VSS	60	VSS	85	VSS
11	PB3/ADM11/A11	36	P96/$\overline{\text{BR}}$	61	$\overline{\text{NMIRQ}}$	86	VDD2(VPP) *
12	PB2/ADM10/A10	37	P95/$\overline{\text{DK}}$	62	P55/SBO3/TM5IO/TM13IO	87	P20/D8
13	PB1/ADM9/A9	38	P94/$\overline{\text{WE1}}$	63	P54/SBI3/TM4IO/TM12IO	88	P17/D7
14	PB0/ADM8/A8	39	P93/$\overline{\text{WE0}}$	64	P53/SBT3/TM3IO/TM11IO	89	P16/D6
15	VDD	40	P92/$\overline{\text{RE}}$	65	P52/SBO2/TM2IO	90	P15/D5
16	PA7/ADM7/A7	41	VDD	66	P51/SBI2/TM1IO	91	P14/D4
17	PA6/ADM6/A6	42	EXMOD1/P91	67	P50/SBT2/TM0IO	92	VDD
18	PA5/ADM5/A5	43	EXMOD0/P90	68	P45/SBO1/$\overline{\text{DWE}}$	93	P13/D3
19	PA4/ADM4/A4	44	AVSS	69	VDD	94	P12/D2
20	PA3/ADM3/A3	45	P83/AN3/IRQ7	70	P44/SBI1/$\overline{\text{DCAS1}}$	95	P11/RWSEL/D1
21	VSS	46	P82/AN2/IRQ6	71	P43/SBT1/$\overline{\text{DCAS0}}$	96	P10/AS/D0
22	PA2/ADM2/A2	47	P81/AN1/IRQ5	72	VSS	97	P02/$\overline{\text{CAS}}$/A22
23	PA1/ADM1/A1	48	P80/AN0/IRQ4	73	P42/SBO0	98	VSS
24	PA0/ADM0/A0	49	VREFH	74	P41/SBI0	99	P01/A21
25	CKSEL	50	AVDD	75	P40/SBT0	100	P00/A20

- Pins for which two or more names are shown are multipurpose pins.

* "VDD2" in the case of the MN103001G, "VPP" in the case of the MN1030F01K.

1.4.2 Pin Functions

Table 1-4-2 shows the function of each pin of this microcontroller.

Table 1-4-2 Pin Function Table (1/2)

Category	Pin name	Input/ Output	Number of pins	Pin Function
Power supply	VDD		8	Digital system power supply (+3.3 V)
	VSS		7	Digital system GND
	VDD2(VPP)		1	"VDD2" in the case of the MN103001G, "VPP" in the case of the MN1030F01K. Connect to 5 V or 3.3 V. Always input 5 V to VPP when writing.
Clock	OSCI	I	1	Oscillator input (use input of 3.3 V to 0 V amplitude only.)
	OSCO	O	1	Oscillator output (Open when using an external clock input)
	SYSCLK	O	1	System clock output (multipurpose)
	CKSEL	I	1	Switch for using/not using PLL
	PVDD		1	PLL power supply (+3.3 V)
	PVSS		1	PLL GND
Address bus	A23 to 0	O	24	Address lines 23 to 0 (multipurpose, A23 also serves as $\overline{\mathrm{CS3}}$.)
Data bus	D15 to 0	I/O	16	Data lines 15 to 0 (multipurpose)
Address/ Data bus	ADM15 to 0	I/O	(16)	Address/Data lines 15 to 0 (A15 to 0 multipurpose)
Bus control signals	MMOD1 to 0	I	2	Mode setting signals
	EXMOD1 to 0	I	2	Extension mode setting (multipurpose)
	$\overline{\mathrm{RE}}$	O	1	Memory read signal (multipurpose)
	$\overline{\mathrm{CS3}}$ to $\overline{0}$	O	4	Chip select signals (multipurpose)
	$\overline{\mathrm{WE1}}$ to $\overline{0}$	O	2	Memory write signals (multipurpose)
	$\overline{\mathrm{DK}}$	I	1	Data acknowledge signal (multipurpose)
	$\overline{\mathrm{BG}}$	O	1	Bus authority release signal (multipurpose)
	$\overline{\mathrm{BR}}$	I	1	Bus authority request signal (multipurpose)
	$\overline{\mathrm{CAS}}$	O	(1)	DRAM CAS signals (for 2$\overline{\mathrm{WE}}$) (multipurpose of A22)
	$\overline{\mathrm{RAS2}}$ to $\overline{1}$	O	(2)	DRAM RAS signal (multipurpose of $\overline{\mathrm{CS2}}$ to $\overline{1}$)
	AS	O	(1)	Address strobe signal (Also serves as D0.)
	RWSEL	O	(1)	Read/write select (Also serves as D1.)
	$\overline{\mathrm{DCAS1}}$ to $\overline{0}$	O	(2)	DRAM CAS signal (for 2 $\overline{\mathrm{CAS}}$)
	$\overline{\mathrm{DWE}}$	O	(1)	DRAM write signal (for 2 $\overline{\mathrm{CAS}}$)

Table 1-4-2 Pin Function Table (2/2)

Category	Pin name	Input/ Output	Number of pins	Pin Function
Reset	$\overline{\text{RST}}$	I	1	Reset input
Interrupts	$\overline{\text{NMIRQ}}$	I	1	External non-maskable interrupt input
	IRQ7 to 0	I	8	External interrupt 7 to 0 inputs (multipurpose)
Serial interface	SBI3 to 0	I	4	Serial 3 to 0 data inputs (multipurpose)
	SBO3 to 0	I/O	4	Serial 3 to 0 data inputs/outputs (multipurpose) (SBO3 is output only.)
	SBT3 to 0	I/O	4	Serial 3 to 0 transfer clock inputs/outputs (multipurpose) (SBT3 is input only.)
8-bit timer	TM7IO to TM0IO	I/O	(8)	PWM output/Toggle output/shared by event count input
16-bit timer	TM10IOA	I/O	(1)	PWM output/capture input (multipurpose of IRQ2)
	TM10IOB	I/O	(1)	PWM output/capture input (multipurpose of IRQ3)
	TM13IO to TM11IO	I/O	(3)	Dual use for event count/toggle output
A/D converter	VREFH	I	1	A/D converter reference voltage input (Use input of AVDD to 0 V only.)
	$\overline{\text{ADTRG}}$	I	(1)	A/D converter trigger conversion input (multipurpose of IRQ3)
	AN3 to 0	I	(4)	A/D converter analog signal inputs (multipurpose of IRQ7 to 4) (Use input of VREFH to 0 V only.)
	AVDD		1	Analog system power supply (+3.3 V)
	AVSS		1	Analog system GND
I/O ports	P02 to P00	O	(3)	Port 0; output port (multipurpose)
	P17 to P10	I/O	(8)	Port 1; input/output port (multipurpose)
	P27 to P20	I/O	(8)	Port 2; input/output port (multipurpose)
	P30	I/O	(1)	Port 3; input/output port (multipurpose)
	P45 to P40	I/O	(6)	Port 4; input/output port (multipurpose)
	P55 to P50	I/O	(6)	Port 5; input/output port (multipurpose)
	P63 to P60	I/O	(4)	Port 6; input/output port (multipurpose)
	P73 to P70	O	(4)	Port 7; output port (multipurpose)
	P83 to P80	I	(4)	Port 8; input port (multipurpose)
	P96, P95, P91, P90	I/O	(4)	Port 96, 95, 91, 90; input/output port (multipurpose)
	P97, P94 to P92	O	(4)	Port 97, 94, 93, 92; output port (multipurpose)
	PA7 to PA0	I/O	(8)	Port A; input/output port (multipurpose)
	PB7 to PB0	I/O	(8)	Port B; input/output port (multipurpose)
	PC3 to PC0	O	(4)	Port C; output port (multipurpose)

Notes:

1. A number that is not enclosed in parentheses in the “Number of pins” column indicates the main pins, while a number enclosed in parentheses indicates multipurpose pins.
2. After the reset condition is released, maintain the $\overline{\text{NMIRQ}}$ pin at the high level until the initialization routine (which sets the stack pointer SP) is completed. If the $\overline{\text{NMIRQ}}$ pin is not used, connect it to VDD via a resistor.

2. CPU

2.1 Basic Specifications of CPU

- Structure

  Load/store architecture
  Data/Address/SP Registers x 9
  (Data registers: 32-bit x 4, Address registers: 32-bit x 4, SP: 32-bit x 1)
  Other Registers
  (PC: 32-bit x 1, PSW: 16-bit x 1, Multiply/divide register: 32-bit x 1,
  Branch target registers: 32-bit x 2)

- Instructions

Number of instructions	: 46
Number of addressing modes	: 6
Basic instruction length	: 1 byte
Code assignment	: 1 byte to 2 bytes (basic part) + 0 byte to 6 bytes (extension)

- Basic performance

Maximum internal operating frequency	: 60 MHz*[1] (external oscillation: 15 MHz) 40 MHz*[2] (external oscillation: 10 MHz)
Minimum instruction execution cycle	: 1 cycle (16.7 ns*[1]/25 ns*[2])
Register-register operations	: 1 cycle
Load/store	: 1 cycle
Conditional branch	: 1 cycle to 3 cycles

- Pipeline

  5-stage (Instruction fetch, decode, execute, memory access, write-back)

- Address space

  4 GB
  Unified space for instructions and data
  (Instructions can not be read from internal data RAM.)

*[1] in the case of the MN103001G.

*[2] in the case of the MN1030F01K.

2.2 Block Diagram

The block diagram for this microcontroller, focusing on the CPU, is shown below.

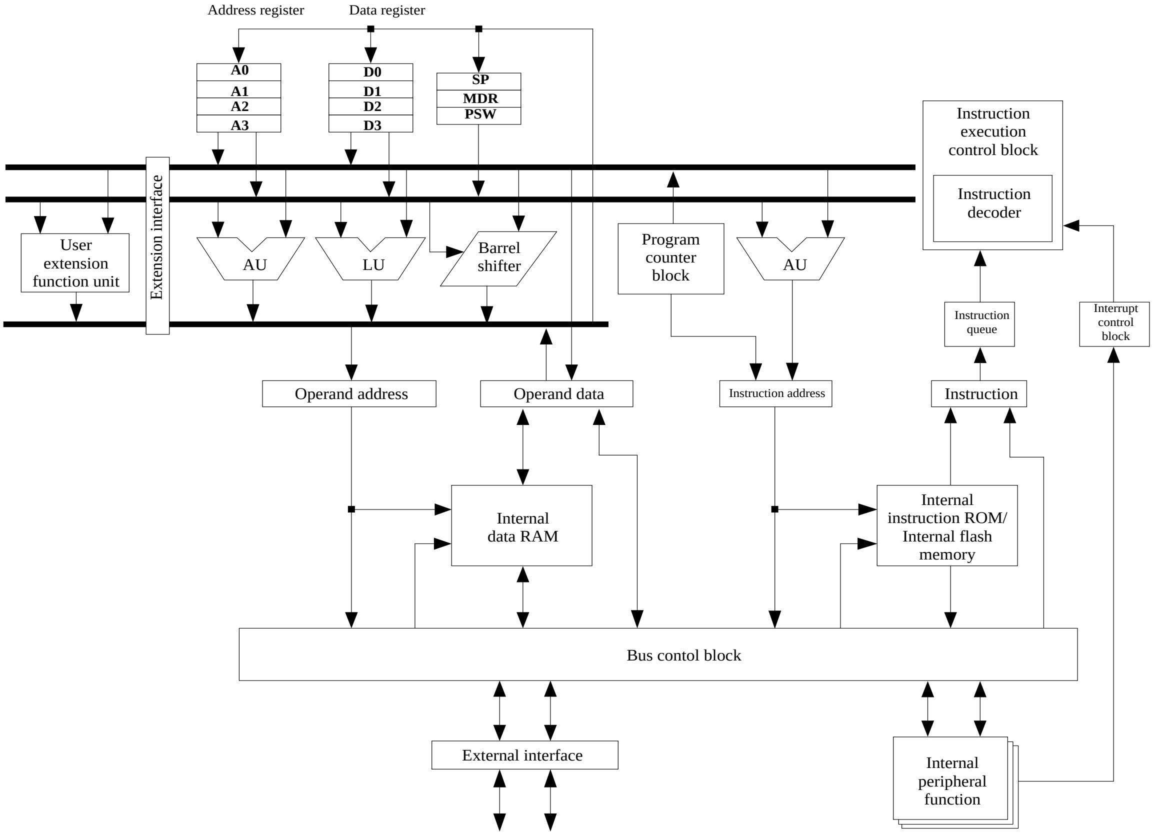


Fig. 2-2-1 CPU Core Block Diagram

2.3 Programming Model

2.3.1 CPU Registers

- The register set is divided into data registers that are used for arithmetic operations, etc., address registers that are used for pointers, and a stack pointer. This arrangement contributes greatly to the improved performance of the internal architecture, through reduction of instruction code size, improved parallelism in pipeline processing, etc.
- This register enables programming in C and other high-level languages.

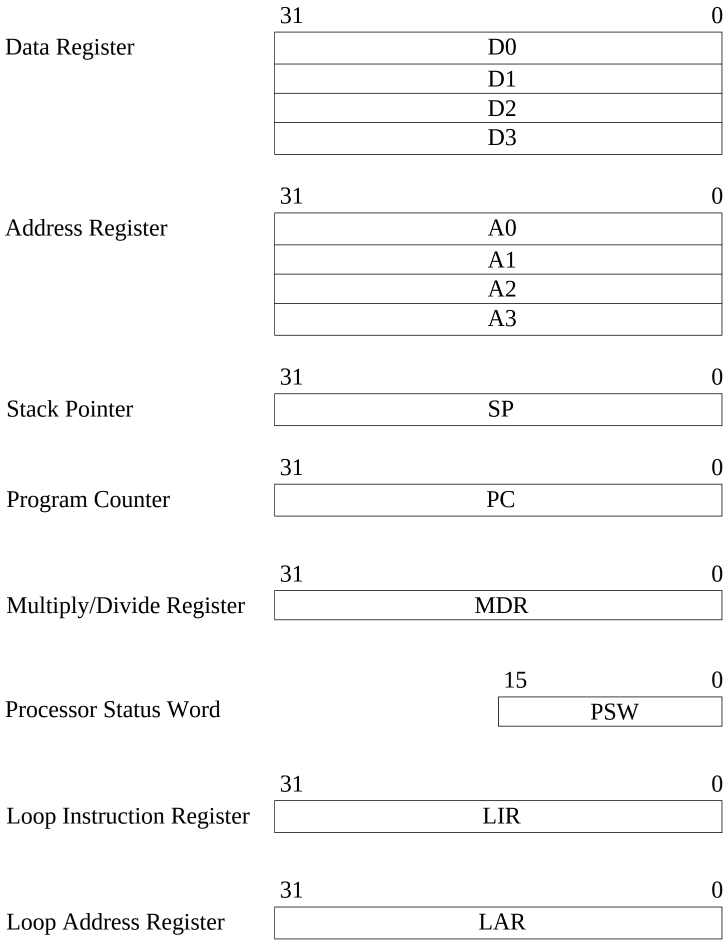


Fig. 2-3-1 CPU Registers

- The loop instruction register (LIR) and the loop address register (LAR) are used to provide high-speed execution of branch instructions. High-speed loop control is performed by loading the branch target instruction and following fetch address with the SETLB instruction and forming the loop using the Lcc instruction.

■ Data Register (32-bit x 4)
This register can be used generally for all operations. Operations are performed with a 32-bit length and the data size is converted when sending data to and from the memory or by executing the EXTB or EXTH instructions. When loading data, 8-bit data is zero-extended to 32 bits and sent to the register. When storing data, the lower 8 bits of the register are sent to the memory. When handling the loaded 8-bit data as a signed integer, the data is sign-extended from 8 bits to 32 bits with the EXTB instruction. When loading data, 16-bit data is zero-extended to 32 bits and sent to the register. When storing data, the lower 16 bits of the register are sent to the memory. When handling the loaded 16-bit data as a signed integer, the data is sign-extended from 16 bits to 32 bits with the EXTH instruction.

■ Address Register (32-bit x 4)
This register is used as an address pointer, and only instructions (addition, subtraction and comparison) for address calculation are supported.
The address register data is used for pointers, and data is normally sent to and from the memory with a 32-bit length.

■ Stack Pointer (32-bit x 1)
This pointer designates the first address of the stack region.

■ Program Counter (32-bit x 1)
This counter designates the address of the command being executed.

■ Multiply/Divide Register (32-bit x 1)
This register is provided for multiply and divide instructions. It holds the upper 32 bits of 64-bit multiplication results for multiply instructions and the remainder (32 bits) for divide instructions. Also, the upper 32 bits of the dividend are loaded to this register before executing divide instructions.

■ Processor Status Word (16-bit x 1)
This register indicates the CPU status, and contains the operation result flags and interrupt mask level, etc.

15															0
0	0	S1	S0	IE	IM2	IM1	IM0	0	0	0	0	V	C	N	Z

Fig. 2-3-2 Processor Status Word

Z: Zero Flag

This flag is set when an operation result is all zeroes, and is cleared by any other result. This flag is also cleared by a reset.

N: Negative Flag

This flag is set if the MSB of an operation result is "1", and is cleared if the MSB is "0". This flag is also cleared by a reset.

C: Carry Flag

This flag is set when a carry or borrow to or from the MSB is generated in the course of executing an operation, and is cleared if no carry or borrow is generated. This flag is also cleared by a reset.

V: Overflow Flag

This flag is set when an overflow occurs in a signed value in the course of executing an operation, and is cleared if no overflow is generated. This flag is also cleared by a reset.

IM2 to IM0: Interrupt Mask

These bits indicate the CPU interrupt mask level. The three bits define the mask level from level 0 (000) to level 7 (111), with level 0 being the highest mask level. The CPU accepts only those interrupt requests of a level higher than the mask level indicated here.

When an interrupt is accepted, the IM bits are set to the priority level of that interrupt. Until the processing of the accepted interrupt is completed, the CPU does not accept interrupts with the same interrupt level or lower.

The interrupt mask level is set to level 0 (000) by a reset.

IE: Interrupt Enable

Setting this bit to “1” allows interrupts to be accepted.

Once the CPU accepts an interrupt request, the IE bit is cleared to "0" and further acceptance of interrupts is prohibited. Accordingly, the IE bit must be reset when processing nested interrupts.

This bit is cleared when the system is reset.

S1 to S0: Software Bits

These are the software control bits for the operating system. These bits cannot be used by general user programs. These bits are cleared by a reset.

For details on changes of these flags, refer to the "Instruction Manual".

■ Loop Instruction Register (32-bit x 1)

This register is provided for the branch instruction (Lcc), and is used to load branch target instructions with the SETLB instruction. This register works together with the Lcc instruction to enable high-speed loop control.

■ Loop Address Register (32-bit x 1)

This register is provided for the branch instruction (Lcc), and is used to load following fetch addresses with the SETLB instruction.

2.3.2 Control Registers

This microcontroller uses the memory-mapped-I/O method and allocates the peripheral circuit registers to the internal I/O space between addresses x'20000000 and x'3FFFFFFF.

The registers listed below are described in this section. For details on other control registers, refer to the respective sections that explain the various internal peripheral functions.

Table 2-3-1 List of Control Registers

Address	Name	Symbol	Number of bits	Initial value	Access size
x'20000000	Interrupt vector register 0	IVAR0	16	x'XXXX	16
x'20000004	Interrupt vector register 1	IVAR1	16	x'XXXX	16
x'20000008	Interrupt vector register 2	IVAR2	16	x'XXXX	16
x'2000000C	Interrupt vector register 3	IVAR3	16	x'XXXX	16
x'20000010	Interrupt vector register 4	IVAR4	16	x'XXXX	16
x'20000014	Interrupt vector register 5	IVAR5	16	x'XXXX	16
x'20000018	Interrupt vector register 6	IVAR6	16	x'XXXX	16
x'20000020	Core's internal memory control register	MEMCTRC	16	x'0007	16
x'20000040	CPU mode register	CPUM	16	x'0000	16

Interrupt Vector Register (IVARn) (n = 0, 1, 2, 3, 4, 5, 6)

The interrupt vector register (IVAR0 to IVAR6) contains the lower 16 bits of the start address of the interrupt handler for interrupts of the level accepted by the CPU. IVAR0 corresponds to level 0 interrupts; in similar fashion, IVAR1 to IVAR6 correspond to levels 1 to 6, respectively. IVAR0 to IVAR6 are allocated to the internal I/O space between addresses x'20000000 to x'20000018, respectively.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	IVAR n15	IVAR n14	IVAR n13	IVAR n12	IVAR n11	IVAR n10	IVAR n9	IVAR n8	IVAR n7	IVAR n6	IVAR n5	IVAR n4	IVAR n3	IVAR n2	IVAR n1	IVAR n0
Reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit No.	Bit name	Description
15 to 0	IVARn15 to 0	Lower 16 bits of the start address of the level interrupt handler

The IVARn register should be accessed by halfwords (16 bits). Byte and word access is not supported.
Note that the upper 16 bits of the start address of the level interrupt handler are fixed to x'4000.

Core's Internal Memory Control Register (MEMCTRC)

The core's internal memory control register (MEMCTRC) sets the number of waits for the memory mounted inside this microcontroller. This register is allocated to the internal I/O space at address x'20000020.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	—	—	—	—	—	—	—	—	—	—	—	—	LD USE	EXT WAIT	DROM W	ROM WAIT
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Writing these bits is prohibited, since operation is guaranteed only with the settings that are in place after a reset.

CPU Mode Register (CPUM)

The CPU mode register (CPUM) sets the clock operating mode for the CPU and peripheral blocks. This register is allocated to the internal I/O space at address x'20000040.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	—	—	—	—	—	—	—	—	—	—	OSCID	STOP	HALT	SLEEP	OSC1	OSC0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit No.	Bit name	Description
0	OSC0	Always returns "0" when read. Always write "0".
1	OSC1	Always returns "0" when read. Always write "0".
2	SLEEP	CPU operating mode control flag (SLEEP transfer request)
3	HALT	CPU operating mode control flag (HALT transfer request)
4	STOP	CPU operating mode control flag (STOP transfer request)
5	OSCID	Always returns "0" when read. Always write "0".
15 to 6	—	reserved

The various operating modes can be set by setting the bits as shown in the table below.

Oscillation control and operating mode control

Operating mode	STOP	HALT	SLEEP	OSC1	OSC0	Clock oscillation	CPU operation clock	Peripheral function operation clock
NORMAL	0	0	0	0	0	Oscillating	Running	Running
HALT	0	1	0	0	0	Oscillating	Stopped	Stopped
SLEEP	0	0	1	0	0	Oscillating	Stopped	Running
STOP	1	0	0	0	0	Stopped	Stopped	Stopped

The CPUM register should be accessed by halfwords (16 bits). Byte and word access is not supported.

If the CPUM register is accessed to make a transition to an operating mode of SLEEP/HALT/STOP during execution of a program in external memory, a branch instruction should not be located within the three instructions immediately following the CPUM register access instruction.

2.4 Instructions

2.4.1 Addressing Modes

The 32-bit microcontroller is equipped with the following 6 addressing modes which are frequently used with compilers.

All 6 addressing modes of register direct, immediate value, register indirect, register indirect with displacement, absolute and register indirect with index can be used with data transfer group instructions.

The 2 addressing modes of register direct and immediate addressing can be used with register operation instructions.

Register indirect with index addressing is an addressing mode used to efficiently access arrays and other data.

Table 2-4-1 Addressing Mode Types

Addressing mode		Address calculation	Effective address
Register direct	Dm / Dn Am / An	——	——
Immediate value	imm8 / regs imm16 imm32 imm40 imm48	——	——
Register indirect	(Am) / (An)	31 0 Am/An	31 0 (32-bit address)
Register indirect with displacement	(d8, Am)/(d8, An) : d8 is sign-extended (d16, Am)/(d16, An) : d16 is sign-extended (d32, Am)/(d32, An)	31 0 Am/An + 31 15 7 0 d32/d16/d8	31 0 (32-bit address)
	(Branch instructions only) (d8, PC) : d8 is sign-extended (d16, PC) : d16 is sign-extended (d32, PC)	31 0 PC + 31 15 7 0 d32/d16/d8	31 0 (32-bit address)
	(d8, SP) : d8 is zero-extended (d16, SP) : d16 is zero-extended (d32, SP)	31 0 SP + 31 15 7 0 d32/d16/d8	31 0 (32-bit address)
Absolute	(abs16) : abs16 is zero-extended (abs32)	31 15 0 abs32/abs16	31 0 (32-bit address)
Register indirect with index	(Di, Am)/(Di, An)	31 0 Am/An + 31 0 Di	31 0 (32-bit address)

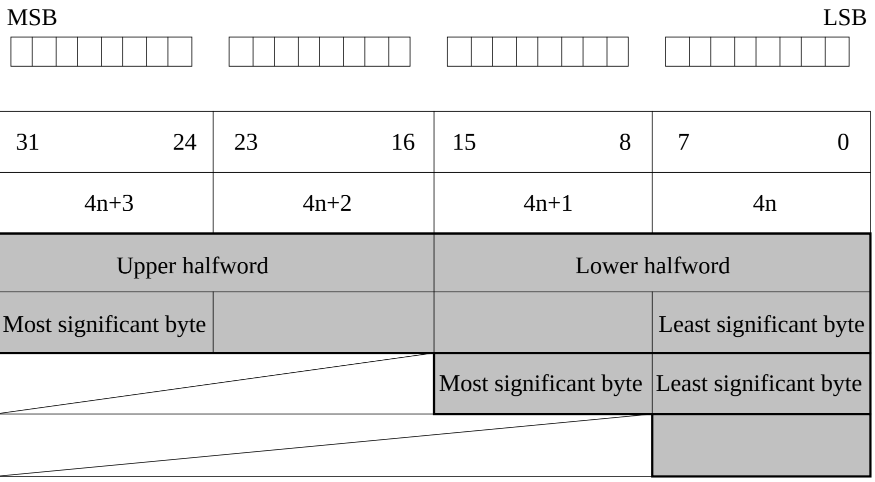
When accessing data using the register indirect with displacement and register indirect with index modes, the base address (the contents of Am, An and SP) and the effective address must be located within the same address space. For details on memory spaces, refer to section 4.1, "Memory Mode Types and Selection."

2.4.2 Data Types

Data types can be processed in the four types of bit, byte, halfword and word data. Byte data, halfword data and word data can be handled as signed and unsigned data. The sign bit is MSB.
The data in the memory must be aligned data. In other words, the two bits on the LSB side of addresses containing word data must be "00" (addresses which are a multiple of 4), and the LSB of addresses containing halfword data must be "0" (addresses which are a multiple of 2).
Byte and bit placement conforms with the Little Endian format. Therefore, the address of the byte data on the MSB side of halfword data is the LSB side byte data address + 1,and the address of the byte data on the MSB side of word data is the LSB side byte data address + 3. The bit number for bit data starts at 0 on the LSB and increases towards the MSB.

Table 2-4-2 Data Types

(1) Bit data	
(2) Byte data Unsigned 8-bit Signed 8-bit	 (sign bit: MSB)
(3) Halfword data Unsigned 16-bit Signed 16-bit	 (sign bit: MSB)
(4) Word data Unsigned 32-bit Signed 32-bit	 (sign bit: MSB)



Bit No.	31 24	23 16	15 8	7 0
Address in the memory	4n+3	4n+2	4n+1	4n
Word data	Upper halfword		Lower halfword	
	Most significant byte			Least significant byte
Halfword data			Most significant byte	Least significant byte
Byte data				

Fig. 2-4-1 Little Endian Format

2.4.3 Instruction Set

The instruction set has a simple organization, and features the generation of compact and optimized code through a C compiler.

The instruction code size is reduced by making the basic instruction word length one byte. As a result, increases in the code size of the assembler program can be kept to a minimum even though the instruction set is simple, with data transfers to and from memory limited to load and store operations.

Table 2-4-3 Instruction Types (All 46 types and extension instructions)

• Transfer instructions		
	MOV	Transfer of word data between registers Transfer of word data between registers and the memory Transfer of immediate values to registers
	MOVBU	Transfer of byte data between registers and the memory (zero-extension)
	MOVHU	Transfer of halfword data between registers and the memory (zero-extension)
	EXT	64-bit sign-extension of word data
	EXTB	32-bit sign-extension of byte data
	EXTBU	32-bit zero-extension of byte data
	EXTH	32-bit sign-extension of halfword data
	EXTHU	32-bit zero-extension of halfword data
	MOVM	Transfer between multiple registers and the memory
	CLR	Data clear
• Arithmetic instructions		
	ADD	Addition
	ADDC	Addition with carry
	SUB	Subtraction
	SUBC	Subtraction with borrow
	MUL	Signed multiplication
	MULU	Unsigned multiplication
	DIV	Signed division
	DIVU	Unsigned division
	INC	Increment by 1
	INC4	Increment by 4
• Compare instructions		
	CMP	Compare
• Logical instructions		
	AND	And
	OR	Inclusive Or
	XOR	Exclusive Or
	NOT	Not (complement of 1)

• Bit instructions		
	BTST	Bit Test
	BSET	Test and set (processing unit: byte)
	BCLR	Test and clear (processing unit: byte)
• Shift instructions		
	ASR	Shift Right Arithmetic
	LSR	Shift Right Logical
	ASL	Shift Left Arithmetic
	ASL2	Shift Left 2-bit Arithmetic
	ROR	Rotate 1 bit to the right
	ROL	Rotate 1 bit to the left
• Branch instructions		
	Bcc	Branch on condition codes (PC relative)
	Lcc	Loop on condition codes (PC relative)
	SETLB	Set loop buffer
	JMP	Unconditional branch (PC relative, register indirect)
	CALL	Subroutine call (Advanced function)
	CALLS	Subroutine call
	RET	Return from subroutine (Advanced function)
	RETF	Return from subroutine (Advanced function, high-speed)
	RETS	Return from subroutine
	RTI	Return from interrupt program
	TRAP	Subroutine call to specified address
	NOP	No operation
• Extension instructions		
	UDF	User extension instruction (sign-extension)
	UDFU	User extension instruction (zero-extension)

Note:

Interrupts are prohibited and the bus is locked (occupied by the CPU) when executing BSET or BCLR, however, if a BSET or BCLR instruction is executed during program execution in external memory, a bus authority release due to an external bus request may be interposed between the data read and data write by the BSET or BCLR instruction. If the atomic bus cycles (i.e. bus lock) of the BSET or BCLR instruction need to be guaranteed in a system that uses multiple processors, either of the following measures should be taken.

1. A program in which a BSET or BCLR instruction is executed should be placed in internal memory.

2. Designate the bus authority request pin ($\overline{\mathrm{BR}}$) as a general-purpose input port, and the bus authority release pin ($\overline{\mathrm{BG}}$) as a general-purpose output port, for instance, so that bus requests cannot be accepted during execution of a BSET or BCLR instruction.

2.5 Interrupts

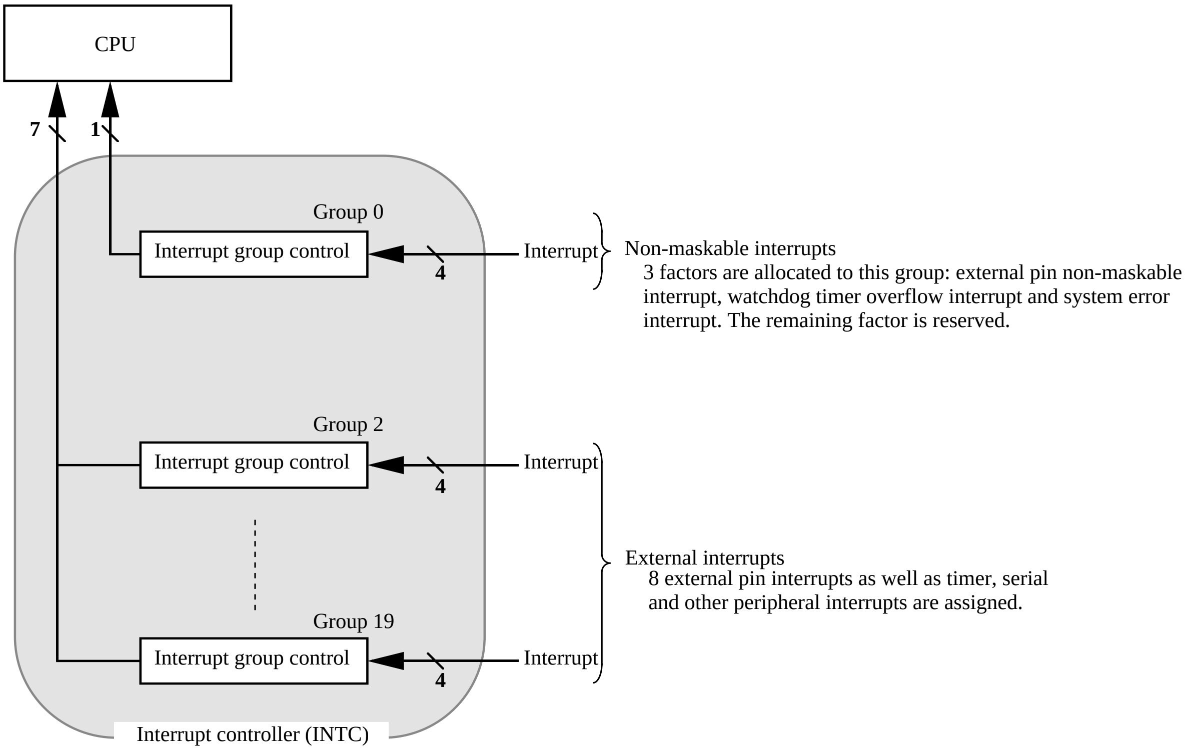
2.5.1 Overview of Interrupts

The most important key to real-time control is the ability to shift quickly to interrupt handler processing.
If an interrupt is generated during the execution of an instruction that requires multiple cycles for execution (multiplication or division instructions, for example), interrupt response is improved by aborting the execution of the instruction and immediately accepting the interrupt. After control returns from the interrupt processing program, the aborted instruction is re-executed.
In addition, by minimizing the resources saved to memory to just the 6 bytes of the PC and the PSW when an interrupt is generated, the speed of interrupt processing is improved, as is the flexibility of software control. Furthermore, fast response and optimal program allocation are possible by placing interrupt processing programs at different addresses for each interrupt level.
This microcontroller has the interrupts shown below. When any of these interrupts occurs, control is shifted to the appropriate processing program in accordance with the cause.

- Reset interrupt
- Non-maskable interrupt
- Level interrupt n (n = 0 to 6)

↑ Priority ranking

Fig. 2-5-1 shows an overview of the interrupt system. This microcontroller is equipped with 19 interrupt group control blocks outside the CPU, and controls the interrupts of each group separately. Each interrupt group control block can accept up to 4 interrupt requests. This allows the controller to support to 38 interrupt factors, providing it with high expandability and enabling flexible ASIC support.
Except for the reset interrupt, all interrupts from the timer and other peripheral circuits and external pin interrupts are registered in the interrupt group control blocks. Then, the interrupt requests which pass the interrupt priority level (level 0 to 6) set in the interrupt group control blocks are output to the CPU. Groups 0 is assigned to non-maskable interrupts only.



Non-maskable interrupts
3 factors are allocated to this group: external pin non-maskable interrupt, watchdog timer overflow interrupt and system error interrupt. The remaining factor is reserved.

External interrupts
8 external pin interrupts as well as timer, serial and other peripheral interrupts are assigned.

Fig. 2-5-1 Overview of the Interrupt System

2.5.2 Registers

[Flags in the PSW] (CPU)

Interrupt-related flags in the processor status word (PSW) include interrupt enable and interrupt mask level.

IE (Interrupt Enable) R/W

- This flag allows all interrupts to be accepted except for non-maskable interrupts and reset interrupts. Interrupts are allowed when IE = 1. IE = 0 when the system is reset.
- When an interrupt is accepted, IE is cleared (interrupt prohibited). Set IE when accepting nested interrupts within the interrupt handler.

IM2 to IM0 (Interrupt Mask Level) R/W

- This holds the current interrupt mask level. When IE = 1, CPU accepts interrupts with levels higher than IM2 to IM0. Level 0 (000) when the system is reset.
- The following table shows the relationship between mask levels and acceptable interrupt levels.

Table 2-5-1 Relationship between Mask Levels and Interrupt Levels that Can Be Accepted

Interrupt mask level			Acceptable interrupt level
IM2	IM1	IM0	
0	0	0	Interrupt prohibited (only non-maskable interrupts accepted)
0	0	1	0
0	1	0	0-1
0	1	1	0-2
1	0	0	0-3
1	0	1	0-4
1	1	0	0-5
1	1	1	0-6

[Interrupt Control Registers (GnICR)] R/W halfword/byte access

Interrupt control registers (GnICR: n = 0, 2 to 19) combine interrupt priority level, interrupt enable, interrupt request and interrupt detect fields into a single register in order to control CPU external peripheral interrupts. There are 19 interrupt control registers, one for each group, and they are located in the internal I/O space from x'34000100 to x'3400014C. Register G0ICR is dedicated for non-maskable interrupts, and G0ICR is called NMICR (from the least significant bit: external pin non-maskable interrupt, watchdog timer overflow interrupt, system error interrupt). Fig. 2-5-2 shows the interrupt control register (GnICR) configuration, and each field is described in detail as follows.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
G0ICR (NMICR)	0	0	0	0	0	0	0	0	0	0	0	0	ID			

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GnICR (n = 2 to 19)	0	LV			IE				IR				ID			

Fig. 2-5-2 Interrupt Control Register (GnICR)

LV2 to LV0 (Interrupt Priority Level) R/W

- This 3-bit field sets the interrupt priority level. When the interrupt priority level set in LV2 to LV0 is higher than the interrupt mask level set in IM2 to IM0 in the PSW (i.e., the value set in LV2 to LV0 is smaller than the value set in IM2 to IM0), interrupts in the corresponding interrupt group are enabled. All interrupts (max. 4) in the same interrupt group have the interrupt priority level specified by LV2 to LV0.
- When interrupt requests are asserted simultaneously from multiple interrupt groups, the group with the highest interrupt priority level is accepted. Also, when multiple interrupt groups are set to the same interrupt priority level, the interrupt from the group with the highest priority ranking (the interrupt group with the smallest group number) is accepted.
- All bits are cleared to "0" when the system is reset.

IE3 to IE0 (Interrupt Enable) R/W

- This field has up to 4 bits which specify interrupt approval. The IE3 to IE0 bits correspond to each interrupt factor (max. 4) in the interrupt group. Interrupts are enabled when the corresponding IE3 to IE0 bit is "1".
- Interrupt occurs when IR3 to IR0 and IE3 to IE0 are set.
- All bits are cleared to "0" when the system is reset.

IR3 to IR0 (Interrupt Request) R/W

- This field has up to 4 bits which register interrupt requests. The IR3 to IR0 bits correspond to each interrupt. After the interrupts are accepted, IR3 to IR0 should be cleared by the software during the interrupt handler.
- All bits are cleared to "0" when the system is reset.
- Conditions for setting and clearing IR3 to IR0 are listed below.

ID3 to ID0 (Interrupt Detect) R/W

- This field has up to 4 bits which contain the logical product of IE3 to IE0 and IR3 to IR0. When an interrupt allowed by IE3 to IE0 occurs, the bit corresponding to that interrupt goes to "1". This field is used to specify interrupts within groups during interrupt processing.
- Interrupt requests are canceled by writing the specified values in IR3 to IR0 and ID3 to ID0 and clearing the interrupt request field.

ID change (G0ICR)

Write ID	ID after write
0	Unchanged
1	0

IR change (GnICR: n = 2 to 19)

Write		IR after write
IR	ID	
0	0	Unchanged
0	1	0
1	0	Unchanged
1	1	1

[Interrupt Accept Group Register (IAGR)] R halfword/byte access

During a register read, the interrupt accept group register (IAGR) indicates the smallest group number of the groups that are generating an interrupt of the interrupt levels accepted by the CPU, which are indicated by IM2 to IM0 of the PSW. This register is allocated to address x'34000200 in the internal I/O space. The GN4 to GN0 field (5 bits) corresponds to the interrupt group number. A branch destination of the interrupt program for each group can be found, for example, by referencing the contents of the address obtained by adding the interrupt accept group register value to the leading address of the interrupt vector table. The interrupt accept group register is a read only register, and writing cannot be performed. When there are no interrupt factors of the applicable interrupt level, IAGR becomes 0.
Accessing IAGR is meaningless during non-maskable interrupts.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IAGR	0	0	0	0	0	0	0	0	0	GN					0	0

Fig. 2-5-3 Interrupt Accept Group Register

[Interrupt Vector Address Register (IVARn)] R/W halfword access

The interrupt vector register (IVAR0 to IVAR6) contains the lower 16 bits of the start address of the interrupt handler for interrupts of the accepted level. This register is allocated between addresses x'20000000 to x'20000018 in the internal I/O space. The start address of interrupt levels 0 to 6 correspond to IVAR0 to IVAR6. When an interrupt occurs, control is transferred to the address which is comprised of the upper 16 bits (x'4000) and the lower 16 bits (IVARn). This register is undefined when the system is reset.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IVARn																

Fig. 2-5-4 Interrupt Vector Address Register

2.5.3 Interrupt Types

The three types of interrupts are listed below:

[Reset interrupt]

The reset interrupt is the interrupt with the highest priority level, and is generated by setting the $\overline{\text{RST}}$ pin to "L" level. As a result of the reset interrupt, the registers, etc., are initialized. When the $\overline{\text{RST}}$ pin goes to "H" level, the microcontroller waits until the oscillation of the internal clock stabilizes, and then begins executing program instructions starting from address x'40000000.

[Non-maskable Interrupts]

Non-maskable interrupts are accepted regardless of the PSW interrupt enable (IE) and interrupt mask level IM2 to IM0 values. These interrupts include external pin non-maskable interrupt, watchdog timer overflow interrupt and system error interrupt.

When a non-maskable interrupt is accepted, control transfers to an interrupt processing program located at x'40000008 or beyond.

The interrupt handler accesses NMICR to analyze the interrupt factor, performs interrupt processing, cancels the interrupt factor, and then returns to the normal program using the RTI instruction.

External pin non-maskable interrupt

External pin non-maskable interrupt is generated when the $\overline{\text{NMIRQ}}$ pin goes to "L" level. If an external pin non-maskable interrupt is generated, the external non-maskable interrupt request flag (NMIF) in the non-maskable interrupt control register (NMICR) is set to "1".

Watchdog timer overflow interrupt

Watchdog timer overflow interrupt occurs when the watchdog timer count operation control flag (WDCNE) in the watchdog timer control register (WDCTR) is "1" and the watchdog timer overflows. If a watchdog interrupt is generated, the watchdog timer overflow interrupt request flag (WDIF) in the non-maskable interrupt control register (NMICR) is set to "1".

System error interrupt

System error interrupt occurs when an unaligned memory access or an unimplemented instruction is executed or other fatal error occurs. If a system error interrupt is generated, the system error interrupt request flag (SYSEF) in the non-maskable interrupt control register (NMICR) is set to "1".

Note: Do not change the interrupt enable (IE) in PSW during non-maskable interrupt processing.

[Level interrupts]

Level interrupts are interrupts for which the interrupt level can be controlled through the interrupt enable (IE) and interrupt mask (IM2 to IM0) bits in the PSW. Level interrupts are interrupts from the interrupt group controllers external to the CPU (in other words, peripheral interrupts). There are 18 groups, or 35 interrupt factors.
Each interrupt group controller includes an interrupt control register (GnICR); the interrupt priority level can be set independently for each interrupt group. It is also possible to set the same interrupt priority level for different interrupt groups. If interrupts of the same priority level are generated simultaneously, the interrupts are accepted in the sequence set by the hardware (the lower the interrupt group number, the higher the priority).
When a level interrupt is accepted, the hardware causes the program to branch to an address with the upper 16 bits being "x'4000" and the lower 16 bits indicated by the interrupt vector address register IVARn corresponding to the interrupt level.
The interrupt handler accesses IAGR to analyze the interrupt group, accesses GnICR (n = 2 to 19) to analyze the interrupt factor, performs interrupt processing, cancels the interrupt factor, and then returns to the normal program using the RTI instruction.

2.5.4 Interrupt Definition

When this microcontroller accepts an interrupt, first the sequences automatically processed by the hardware are executed. Then control transfers to interrupt handler by the software and the interrupt handler is started up.
The interrupt processing sequences are described below.

(Interrupt processing sequences executed by the hardware)

1. The PSW is saved to the stack (SP-8).
2. The PC (return address) is saved to the stack (SP-4).
3. The PSW is updated.
   IE is cleared and the accepted interrupt level is set in IM2 to IM0. (IM2 to IM0 is undefined in case of non-maskable interrupts.)
4. The stack pointer is updated. (SP-8 → SP)
5. Control is transferred to the address corresponding to the accepted interrupt factor or the address comprised of the interrupt vector address register (IVARn).

When an interrupt other than a reset interrupt is accepted, control is transferred to the address corresponding to the interrupt factor or the address comprised of the interrupt vector address register. The processing listed below is then performed at the branch destination in order to judge the interrupt factor in further detail.
See "2.5.3 Interrupt Types" for processing reset interrupts.
(Note) In General, Branch instructions (JMP instruction, etc.) are placed at the branch destination for reset interrupts, then it branches to the initialization program.

(Example of pre-processing by the interrupt handler)

1. The registers are saved.
   The saved registers are those used by the interrupt handler.
2. The interrupt group analysis is executed.
2.1 The interrupt acknowledge sequence is executed.
   Interrupt acknowledge consists of reading out the interrupt accept group register (IAGR) to obtain the group number of the interrupt group with the highest priority among the specified interrupt levels.
2.2 The leading address of the interrupt handler for each level is generated.
2.3 Control is transferred to the interrupt handler for each level.
3. When there are multiple factors within the same group, the interrupt control register (GnICR) is read out to designate the factor.
   * In case of non-maskable interrupts, the factor is specified by accessing the NMICR directly without accessing the IAGR.
4. Control is transferred to the interrupt handler for each factor.
   Note that because this microcontroller uses a store buffer when writing data via the bus controller, it is necessary, when releasing the interrupt factor, to read the appropriate register immediately after clearing the interrupt factor in order to wait for the factor in the GnICR to be cleared completely.

(Example of post-processing by the interrupt handler)

5. The registers are restored.
   The restored registers are those saved by the pre-processing.
6. The RTI instruction is executed and control returns to the program before the interrupt.

Fig. 2-5-5 shows the interrupt sequence flow. (when not accepting nested interrupts)
The numbers in the figure correspond to the numbers of processing performed by the interrupt handler in the previous section.

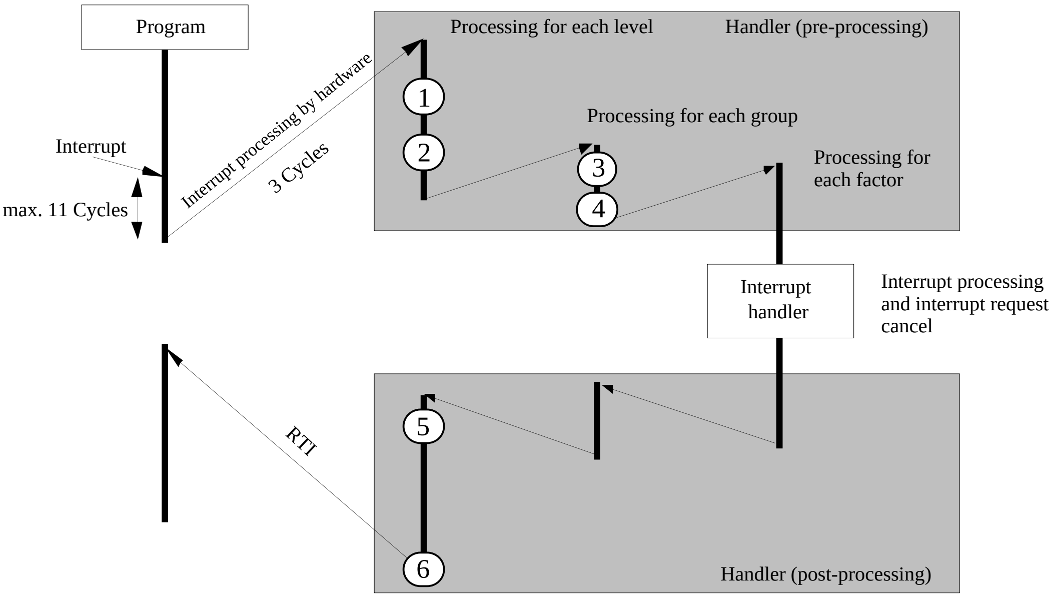

Program
Handler (pre-processing)
Processing for each level
Processing for each group
Processing for each factor
Interrupt
max. 11 Cycles
Interrupt processing by hardware
3 Cycles
1
2
3
4
Interrupt handler
Interrupt processing and interrupt request cancel
RTI
5
6
Handler (post-processing)

Fig. 2-5-5 Interrupt Sequence Flow

An even higher interrupt response speed can be realized by assigning only one factor or only a few factors to a single interrupt level.
Fig. 2-5-6 shows the interrupt sequence flow when assigning one factor to each interrupt level.

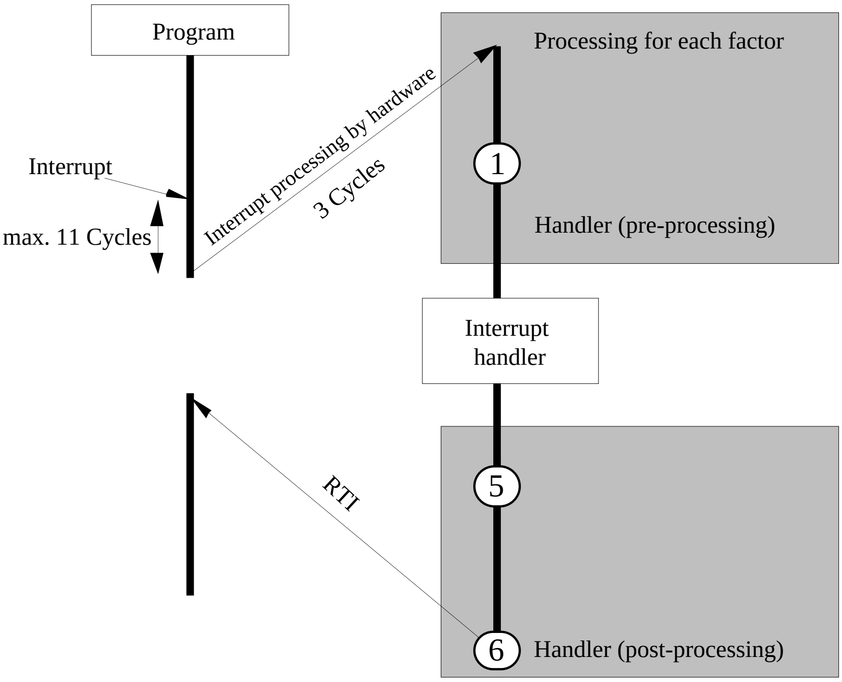


Fig. 2-5-6 Interrupt Sequence Flow

[Nested Interrupts]

When a level interrupt occurs, nested interrupts can be prohibited by clearing IE of the PSW. However, nested interrupts can be achieved even while processing level interrupts by setting IE to "1" during processing. However, in order for nested interrupts to occur, the interrupts must have a higher priority than interrupt mask level IM2 to IM0 of the PSW at that time. (The GnICR interrupt priority level LV2 to LV0 is smaller than the PSW interrupt mask level IM2 to IM0.)
When non-maskable interrupts occur, nesting of level interrupts and non-maskable interrupts is prohibited until the interrupt handler is finished by execution of the RTI instruction.

[Interrupt Acceptance Timing]

If an interrupt request occurs part-way through the execution of an instruction, even instructions which require multiple execution cycles such as multiply/divide and other instructions are aborted if possible and the interrupt is accepted. The aborted instruction is executed again after returning from interrupt processing. Aborting these instructions sets the interrupt acceptance prohibited interval to 11 cycles or less. (The maximum interrupt prohibited interval of 11 cycles occurs when saving or restoring all registers with the MOVM, CALL or RET instructions. This occurs only for special cases such as task context switching.)

[Stack Frame]

When an interrupt is accepted, a stack frame is allocated and the total 6 bytes of information in the PC and PSW are saved in order to return from the interrupt. However, since the transfer of data across the 32-bit boundary is prohibited, the SP value must constantly be set to a multiple of 4. Accordingly, a stack frame is allocated as shown in Fig. 2-5-7 so that the SP value is constantly set to a multiple of 4. Ultimately, an 8-byte area with a total of 6 bytes of information is saved.

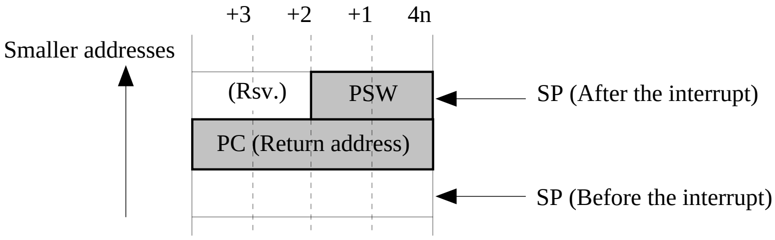


Fig. 2-5-7   Stack Frame Configuration

3. Extension Instruction Specifications

3.1 Operation Extension Function

The MN1030 series 32-bit microcontrollers are provided with 32 extension instructions which can be defined by users. This allows the desired processing to be performed at high speed for each model expansion by assigning multiply, multiply-accumulate, saturation and other application-oriented operations to extension instructions and connecting extension function unit via the extension operation interface of the CPU core.
Extension instructions include instructions UDF00 to UDF15 which transfer register or immediate values to the extension function unit and load the operation results to the data register, and instructions UDF20 to UDF35 which only transfer register to the extension function unit. Processing which performs user-defined operations is assigned to instructions UDF00 to UDF15, and processing which only transfers data to the extension function unit is assigned to instructions UDF20 to UDF35. Extension operations which require three or more inputs can be realized by transferring the input data to the extension function unit beforehand using instructions UDF20 to UDF35 and then performing the operation using instructions UDF00 to UDF15.
The block diagram showing extension function unit connected to the CPU for this series is as follows.
This microcontroller has a 32 x 16 multiplier, priority encoder, and saturation compensation unit on chip. The extension functions that use the extension function unit are explained in section 3.2, "Extension Instructions."

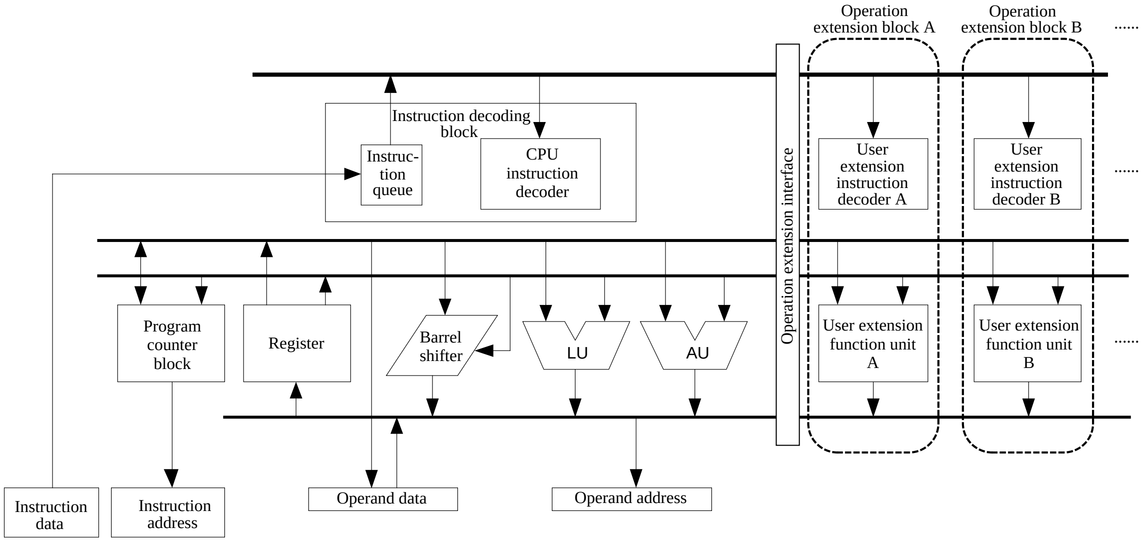

Operation extension block A
Operation extension block B
......
Instruction decoding block
Instruc-tion queue
CPU instruction decoder
Operation extension interface
User extension instruction decoder A
User extension instruction decoder B
......
Program counter block
Register
Barrel shifter
LU
AU
User extension function unit A
User extension function unit B
......
Instruction data
Instruction address
Operand data
Operand address

Fig. 3-1-1 Block Diagram of the Extension Function Unit

3.2 Extension Instructions

3.2.1 Explanation of Notations

The notations used to describe instruction manual are shown below.

OP:	Opcode
Am, An:	Address Register (m, n = 3 to 0)
Dm, Dn:	Data Register (m, n = 3 to 0)
SP:	Stack Pointer
imm:	Immediate value (used as the general meaning)
imm8:	8-bit immediate value
imm16:	16-bit immediate value
imm32:	32-bit immediate value
d8:	8-bit displacement
d16:	16-bit displacement
d32:	32-bit displacement
abs16:	16-bit absolute
abs32:	32-bit absolute
MDR:	Multiply/Divide Register (core built in)
MDRQ:	High-speed multiplication register (inside Extension Function Unit)
LIR:	Loop Instruction Register
LAR:	Loop Address Register
PSW:	Processor Status Word
PC:	Program Counter
( ):	Indirect addressing See "2.4.1 Addressing Modes" for a detailed description.
regs:	Multiple register operand
0x....:	Hexadecimal notation (The numbers following 0x are expressed in hexadecimal notation.)

Notations used to express flag changes are listed below.
("Flag" is the general term used to refer to the lower 4 bits (V, C, N, Z) in the PSW.)

–:	No flag change
+:	Flag change
*:	Undefined
0:	Reset
1:	Set

3.2.2 Extension Block Register Set

The extension block has the following dedicated registers in which it stores the results of high-speed multiplication operations and multiply-and-accumulate operations.

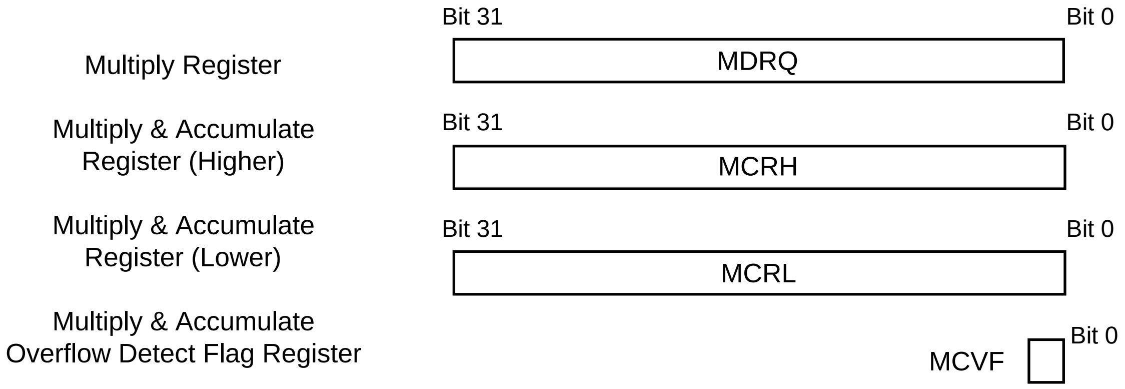

Bit 31
Bit 0
Multiply Register
MDRQ
Multiply & Accumulate Register (Higher)
Bit 31
Bit 0
MCRH
Multiply & Accumulate Register (Lower)
Bit 31
Bit 0
MCRL
Multiply & Accumulate Overflow Detect Flag Register
Bit 0
MCVF

Fig. 3-2-1 Extension Block Register Set

■ Multiply register (32 bits x 1 register)

This register is provided for high-speed multiplication instructions. A multiplication instruction uses this register to store the high-order 32 bits of the 64-bit multiplication result.

■ Multiply-and-accumulate register (higher) (32 bits x 1 register)

This register is provided for multiply-and-accumulate operation instructions. A multiply-and-accumulate operation instruction uses this register to store the high-order 32 bits of the 64-bit multiply-and-accumulate operation result.

■ Multiply-and-accumulate register (lower) (32 bits x 1 register)

This register is provided for multiply-and-accumulate operation instructions. A multiply-and-accumulate operation instruction uses this register to store the low-order 32 bits of the 64-bit multiply-and-accumulate operation result.

■ Multiply-and-accumulate overflow detect flag register (1 bit x 1 register)

This one-bit register is set when an overflow occurs in a multiply-and-accumulate operation. This flag is not cleared until the next CLRMAC instruction or PUTCX instruction is executed.

3.2.3 Extension Instruction Details

PUTX (Register transfer instruction for high-speed multiplication: Load)

[Instruction Format (Macro Name)]

PUTX Dm

[Assembler Mnemonic]

udf20 Dm, Dm

[Operation]

The contents of Dm are transferred to the high-speed multiply register MDRQ.

[Flag Changes]

Flag	Change	Condition
V	–	
C	–	
N	–	
Z	–	

[Programming Cautions]

When "udf20 Dm, Dn" is operated, Dn is ignored.

PUTCX (Register transfer instruction for multiply-and-accumulate operation: Load)

[Instruction Format (Macro Name)]

PUTCX Dm, Dn

[Assembler Mnemonic]

udf21 Dm, Dn

[Operation]

This instruction transfers the contents of Dm to the multiply-and-accumulate register MCRH.
This instruction also transfers the contents of Dn to the multiply-and-accumulate register MCRL.
The contents of the V flag are set in the multiply-and-accumulate overflow detect register MCVF.

[Flag Changes]

Flag	Change	Condition
V	–	
C	–	
N	–	
Z	–	

GETX (Register transfer instruction for high-speed multiplication: Store)

[Instruction Format (Macro Name)]

GETX Dn

[Assembler Mnemonic]

udf15 Dn, Dn

[Operation]

The contents of the high-speed multiply register MDRQ are transferred to Dn.

[Flag Changes]

Flag	Change	Condition
V	0	Always 0
C	0	Always 0
N	+	1 when MSB of the transfer results is 1; 0 in all other cases
Z	+	1 when the transfer results are 0; 0 in all other cases

[Programming Cautions]

There is a one-instruction delay in the updating of the PSW to reflect flag changes.
However, the Bcc and Lcc instructions can evaluate the flags without waiting for the flag changes to be reflected in the PSW.
When "udf15 Dm, Dn" is operated, Dm is ignored.
The operations of "udf15 imm8, Dn", "udf15 imm16, Dn" and "udf15 imm32, Dn" are not assured. In addition, a system error interrupt does not occur in these cases.

GETCHX (Register high-order 32-bit transfer instruction for multiply-and-accumulate operation: Store)

[Instruction Format (Macro Name)]

GETCHX Dn

[Assembler Mnemonic]

udf12 Dn, Dn

[Operation]

This instruction transfers the contents of the multiply-and-accumulate register MCRH to Dn.
The content of the multiply-and-accumulate overflow detect register MCVF is set in the V flag.

[Flag Changes]

When multiply-and-accumulate operation overflow was not detected (MCVF = 0)

Flag	Change	Condition
V	0	Indicates that the multiply-and-accumulate operation is valid.
C	0	Always 0
N	*	Undefined
Z	*	Undefined

When multiply-and-accumulate operation overflow was detected (MCVF = 1)

Flag	Change	Condition
V	1	Indicates that the multiply-and-accumulate operation is invalid.
C	0	Always 0
N	*	Undefined
Z	*	Undefined

[Programming Cautions]

There is a one-instruction delay in the updating of the PSW to reflect flag changes.
However, the Bcc and Lcc instructions can evaluate the flags without waiting for the flag changes to be reflected in the PSW.
When "udf12 Dm, Dn" is operated, Dm is ignored.
The operations of "udf12 imm8, Dn", "udf12 imm16, Dn" and "udf12 imm32, Dn" are not assured. In addition, a system error interrupt does not occur in these cases.

GETCLX (Register low-order 32-bit transfer instruction for multiply-and-accumulate operation: Store)

[Instruction Format (Macro Name)]

GETCLX Dn

[Assembler Mnemonic]

udf13 Dn, Dn

[Operation]

This instruction transfers the contents of the multiply-and-accumulate register MCRL to Dn.
The contents of the multiply-and-accumulate overflow detect register MCVF are set in the V flag.

[Flag Changes]

When multiply-and-accumulate operation overflow was not detected (MCVF = 0)

Flag	Change	Condition
V	0	Indicates that the multiply-and-accumulate operation is valid.
C	0	Always 0
N	*	Undefined
Z	*	Undefined

When multiply-and-accumulate operation overflow was detected (MCVF = 1)

Flag	Change	Condition
V	1	Indicates that the multiply-and-accumulate operation is invalid.
C	0	Always 0
N	*	Undefined
Z	*	Undefined

[Programming Cautions]

There is a one-instruction delay in the updating of the PSW to reflect flag changes.
However, the Bcc and Lcc instructions can evaluate the flags without waiting for the flag changes to be reflected in the PSW.
When "udf13 Dm, Dn" is operated, Dm is ignored.
The operations of "udf13 imm8, Dn", "udf13 imm16, Dn" and "udf13 imm32, Dn" are not assured. In addition, a system error interrupt does not occur in these cases.

CLRMAC (Register clear instruction for multiply-and-accumulate operation)

[Instruction Format (Macro Name)]

CLRMAC

[Assembler Mnemonic]

udf22 D0, D0

[Operation]

This instruction clears the contents of the multiply-and-accumulate registers MCRH and MCRL.
This instruction also clears the contents of the multiply-and-accumulate overflow detect register MCVF.

[Flag Changes]

Flag	Change	Condition
V	–	
C	–	
N	–	
Z	–	

[Programming Cautions]

When "udf22 Dm, Dn" is operated, Dm and Dn are ignored.

MULQ (Signed high-speed multiplication instruction: between registers)

[Instruction Format (Macro Name)]

MULQ Dm, Dn

[Assembler Mnemonic]

udf00 Dm, Dn

[Operation]

This instruction performs multiplication quickly using the multiplier of the extension function unit.
The contents of Dm (signed 32-bit integer: multiplicand) and Dn (signed 32-bit integer: multiplier) are multiplied, and the upper 32 bits of the results (64 bits) are written into the high-speed multiply register MDRQ and the lower 32 bits into Dn.
The significant value range of the multiplicand stored in Dm before the operation is judged (starting point: LSB, judgment unit: 2 bytes), and the operation is only performed for the range containing these significant values. In other words, the smaller the absolute value of the contents stored in Dm, the quicker operation results can be obtained.

[Flag Changes]

Flag	Change	Condition
V	*	Undefined
C	*	Undefined
N	+	1 when MSB of the lower 32 bits of the results is 1; 0 in all other cases
Z	+	1 when the lower 32 bits of results are 0; 0 in all other cases

[Programming Cautions]

PSW updating by flag changes is delayed by one instruction.
However, Bcc and Lcc instructions can evaluate flags without waiting for flag reflection to PSW.

MULQI (Signed high-speed multiplication instruction: between immediate value and register)

[Instruction Format (Macro Name)]

MULQI imm, Dn

[Assembler Mnemonic]

udf00 imm8, Dn :imm8 is sign-extended
udf00 imm16, Dn :imm16 is sign-extended
udf00 imm32, Dn

[Operation]

This instruction performs multiplication quickly using the multiplier of the extension function unit.
The 32-bit data obtained by sign-extending imm (multiplicand) and the contents of Dn (signed 32-bit integer: multiplier) are multiplied, and the upper 32 bits of the results (64 bits) are written into the high-speed multiply register MDRQ and the lower 32 bits into Dn.
The significant value range of the multiplicand stored in imm before the operation is judged (starting point: LSB, judgment unit: 2 bytes), and the operation is only performed for the range containing these significant values. In other words, if the number of imm bits is “16” or less, the operation results will be derived faster.

[Flag Changes]

Flag	Change	Condition
V	*	Undefined
C	*	Undefined
N	+	1 when MSB of the lower 32 bits of the results is 1; 0 in all other cases
Z	+	1 when the lower 32 bits of results are 0; 0 in all other cases

[Programming Cautions]

PSW updating by flag changes is delayed by one instruction.
However, Bcc and Lcc instructions can evaluate flags without waiting for flag reflection to PSW.

MULQU (Unsigned high-speed multiplication instruction: between registers)

[Instruction Format (Macro Name)]

MULQU Dm, Dn

[Assembler Mnemonic]

udf01 Dm, Dn

[Operation]

This instruction performs multiplication quickly using the multiplier of the extension function unit.
The contents of Dm (unsigned 32-bit integer: multiplicand) and Dn (unsigned 32-bit integer: multiplier) are multiplied, and the upper 32 bits of the results (64 bits) are written into the high-speed multiply register MDRQ and the lower 32 bits into Dn.
The significant value range of the multiplicand stored in Dm before the operation is judged (starting point: LSB, judgment unit: 2 bytes), and the operation is only performed for the range containing these significant values. In other words, the smaller the contents stored in Dm, the quicker operation results can be obtained.

[Flag Changes]

Flag	Change	Condition
V	*	Undefined
C	*	Undefined
N	+	1 when MSB of the lower 32 bits of the results is 1; 0 in all other cases
Z	+	1 when the lower 32 bits of results are 0; 0 in all other cases

[Programming Cautions]

PSW updating by flag changes is delayed by one instruction.
However, Bcc and Lcc instructions can evaluate flags without waiting for flag reflection to PSW.

MULQIU (Unsigned high-speed multiplication instruction: between immediate value and register)

[Instruction Format (Macro Name)]

MULQIU imm, Dn

[Assembler Mnemonic]

udfu01 imm8, Dn :imm8 is zero-extended
udfu01 imm16, Dn :imm16 is zero-extended
udfu01 imm32, Dn

[Operation]

This instruction performs multiplication quickly using the multiplier of the extension function unit.
The 32-bit data obtained by zero-extending imm (multiplicand) and the contents of Dn (unsigned 32-bit integer: multiplier) are multiplied, and the upper 32 bits of the results (64 bits) are written into the high-speed multiply register MDRQ and the lower 32 bits into Dn.
The significant value range of the multiplicand stored in imm before the operation is judged (starting point: LSB, judgment unit: 2 bytes), and the operation is only performed for the range containing these significant values. In other words, if the number of imm bits is “16” or less, the operation results will be derived faster.

[Flag Changes]

Flag	Change	Condition
V	*	Undefined
C	*	Undefined
N	+	1 when MSB of the lower 32 bits of the results is 1; 0 in all other cases
Z	+	1 when the lower 32 bits of results are 0; 0 in all other cases

[Programming Cautions]

PSW updating by flag changes is delayed by one instruction.
However, Bcc and Lcc instructions can evaluate flags without waiting for flag reflection to PSW.

MAC (Signed multiply-and-accumulate operation instruction: between registers)

[Instruction Format (Macro Name)]

MAC Dm, Dn

[Assembler Mnemonic]

udf28 Dm, Dn

[Operation]

This instruction performs the multiply-and-accumulate operation using the multiplier and adder in the extension function unit.
The instruction multiplies the contents of Dm (signed 32-bit integer: multiplicand) by the contents of Dn (signed 32-bit integer: multiplier), it adds the product obtained by this multiplication to the cumulative sum (64 bits) of the upper 32 bits and lower 32 bits stored in the respective multiply-and-accumulate registers MCRH and MCRL, and it then stores the upper 32 bits of the result (64 bits) in the multiply-and-accumulate register MCRH and the lower 32 bits in the multiply-and-accumulate register MCRL.
If an overflow from the 64-bit cumulative sum data is generated when the product is added to the cumulative sum, multiply-and-accumulate overflow detection flag 1 is output to register MCVF.

[Flag Changes]

Flag	Change	Condition
V	–	
C	–	
N	–	
Z	–	

[Programming Cautions]

A non-extension instruction that consumes at least two cycles must be inserted between this instruction and the next extension instruction.

MACH (Signed half word data multiply-and-accumulate operation instruction: between registers)

[Instruction Format (Macro Name)]

MACH Dm, Dn

[Assembler Mnemonic]

udf30 Dm, Dn

[Operation]

This instruction performs the multiply-and-accumulate operation using the multiplier and adder in the extension function unit.

The instruction multiplies the contents of Dm (signed 16-bit integer: multiplicand) by the contents of Dn (signed 16-bit integer: multiplier), it adds the product obtained by this multiplication to the cumulative sum (64 bits) of the upper 32 bits and lower 32 bits stored in the respective multiply-and-accumulate registers MCRH and MCRL, and it then stores the upper 32 bits of the result (64 bits) in the multiply-and-accumulate register MCRH and the lower 32 bits in the multiply-and-accumulate register MCRL.

If an overflow from the 64-bit cumulative sum data is generated when the product is added to the cumulative sum, multiply-and-accumulate overflow detection flag 1 is output to register MCVF.

[Flag Changes]

Flag	Change	Condition
V	–	
C	–	
N	–	
Z	–	

[Programming Cautions]

A non-extension instruction that consumes at least one cycle must be inserted between this instruction and the next extension instruction.

MACB (Signed byte data multiply-and-accumulate operation instruction: between registers)

[Instruction Format (Macro Name)]

MACB Dm, Dn

[Assembler Mnemonic]

udf32 Dm, Dn

[Operation]

This instruction performs the multiply-and-accumulate operation using the multiplier and adder in the extension function unit.
The instruction multiplies the contents of Dm (signed 8-bit integer: multiplicand) by the contents of Dn (signed 8-bit integer: multiplier), adds the resulting product to the 32-bit cumulative sum that is stored in the multiply-and-accumulate register MCRL, and then stores the new resulting 32-bit cumulative sum back in multiply-and-accumulate register MCRL.
If an overflow from the 32-bit cumulative sum data is generated when the product is added to the cumulative sum, multiply-and-accumulate overflow detection flag 1 is output to register MCVF.

[Flag Changes]

Flag	Change	Condition
V	–	
C	–	
N	–	
Z	–	

[Programming Cautions]

A non-extension instruction that consumes at least one cycle must be inserted between this instruction and the next extension instruction.

MACU (Unsigned multiply-and-accumulate operation instruction: between registers)

[Instruction Format (Macro Name)]

MACU Dm, Dn

[Assembler Mnemonic]

udf29 Dm, Dn

[Operation]

This instruction performs the multiply-and-accumulate operation using the multiplier and adder in the extension function unit.
The instruction multiplies the contents of Dm (unsigned 32-bit integer: multiplicand) by the contents of Dn (unsigned 32-bit integer: multiplier), it adds the product obtained by this multiplication to the cumulative sum (64 bits) of the upper 32 bits and lower 32 bits stored in the respective multiply-and-accumulate registers MCRH and MCRL, and it then stores the upper 32 bits of the result (64 bits) in the multiply-and-accumulate register MCRH and the lower 32 bits in the multiply-and-accumulate register MCRL.
If an overflow from the 64-bit cumulative sum data is generated when the product is added to the cumulative sum, multiply-and-accumulate overflow detection flag 1 is output to register MCVF.

[Flag Changes]

Flag	Change	Condition
V	–	
C	–	
N	–	
Z	–	

[Programming Cautions]

A non-extension instruction that consumes at least two cycles must be inserted between this instruction and the next extension instruction.

MACHU (Unsigned half word data multiply-and-accumulate operation instruction: between registers)

[Instruction Format (Macro Name)]

MACHU Dm, Dn

[Assembler Mnemonic]

udf31 Dm, Dn

[Operation]

This instruction performs the multiply-and-accumulate operation using the multiplier and adder in the extension function unit.

The instruction multiplies the contents of Dm (unsigned 16-bit integer: multiplicand) by the contents of Dn (unsigned 16-bit integer: multiplier), it adds the product obtained by this multiplication to the cumulative sum (64 bits) of the upper 32 bits and lower 32 bits stored in the respective multiply-and-accumulate registers MCRH and MCRL, and it then stores the upper 32 bits of the result (64 bits) in the multiply-and-accumulate register MCRH and the lower 32 bits in the multiply-and-accumulate register MCRL.

If an overflow from the 64-bit cumulative sum data is generated when the product is added to the cumulative sum, multiply-and-accumulate overflow detection flag 1 is output to register MCVF.

[Flag Changes]

Flag	Change	Condition
V	–	
C	–	
N	–	
Z	–	

[Programming Cautions]

A non-extension instruction that consumes at least one cycle must be inserted between this instruction and the next extension instruction.

MACBU (Unsigned byte data multiply-and-accumulate operation instruction: between registers)

[Instruction Format (Macro Name)]

MACBU   Dm, Dn

[Assembler Mnemonic]

udf33   Dm, Dn

[Operation]

This instruction performs the multiply-and-accumulate operation using the multiplier and adder in the extension function unit.
The instruction multiplies the contents of Dm (unsigned 8-bit integer: multiplicand) by the contents of Dn (unsigned 8-bit integer: multiplier), adds the resulting product to the 32-bit cumulative sum that is stored in the multiply-and-accumulate register MCRL, and then stores the new resulting 32-bit cumulative sum back in multiply-and-accumulate register MCRL.
If an overflow from the 32-bit cumulative sum data is generated when the product is added to the cumulative sum, multiply-and-accumulate overflow detection flag 1 is output to register MCVF.

[Flag Changes]

Flag	Change	Condition
V	–	
C	–	
N	–	
Z	–	

[Programming Cautions]

A non-extension instruction that consumes at least one cycle must be inserted between this instruction and the next extension instruction.

SAT16 (16-bit saturation operation instruction)

[Instruction Format (Macro Name)]

SAT16 Dm, Dn

[Assembler Mnemonic]

udf04 Dm, Dn

[Operation]

When Dm is a 16-bit signed number which is the maximum positive value (0x00007fff) or more, the maximum positive value (0x00007fff) is written into Dn. When Dm is a 16-bit signed number which is the maximum negative value (0xffff8000) or less, the maximum negative value (0xffff8000) is stored in Dn. In all other cases, the contents of Dm are written into Dn.

[Flag Changes]

Flag	Change	Condition
V	*	Undefined
C	*	Undefined
N	+	1 when MSB of the operation results is 1; 0 in all other cases
Z	+	1 when the operation results are 0; 0 in all other cases

[Programming Cautions]

PSW updating by flag changes is delayed by one instruction.
However, Bcc and Lcc instructions can evaluate flags without waiting for flag reflection to PSW.
The operations of "udf04 imm8, Dn", "udf04 imm16, Dn" and "udf04 imm32, Dn" are not assured. In addition, a system error interrupt does not occur in these cases.

SAT24 (24-bit saturation operation instruction)

[Instruction Format (Macro Name)]

SAT24 Dm, Dn

[Assembler Mnemonic]

udf05 Dm, Dn

[Operation]

When Dm is a 24-bit signed number which is the maximum positive value (0x007fffff) or more, the maximum positive value (0x007fffff) is written into Dn. When Dm is a 24-bit signed number which is the maximum negative value (0xff800000) or less, the maximum negative value (0xff800000) is written into Dn. In all other cases, the contents of Dm are written into Dn.

[Flag Changes]

Flag	Change	Condition
V	*	Undefined
C	*	Undefined
N	+	1 when MSB of the operation results is 1; 0 in all other cases
Z	+	1 when the operation results are 0; 0 in all other cases

[Programming Cautions]

PSW updating by flag changes is delayed by one instruction.
However, Bcc and Lcc instructions can evaluate flags without waiting for flag reflection to PSW.
The operations of "udf05 imm8, Dn", "udf05 imm16, Dn" and "udf05 imm32, Dn" are not assured. In addition, a system error interrupt does not occur in these cases.

MCST (Multiply-and-accumulate operation results 8-, 16-, 32-bit saturation operation instruction)

[Instruction Format (Macro Name)]

MCST Dm, Dn
MCST imm8, Dn

[Assembler Mnemonic]

udf02 Dm, Dn
udf02 imm8, Dn : Only 0x20, 0x10, and 0x08 are valid as values for imm8

[Operation]

This instruction sets the contents of the multiply-and-accumulate operation overflow detect register MCVF in the V flag. In addition, depending on the value of Dm or imm8, the following operations are performed.

(1) When the value of Dm or imm8 is 32 (0x00000020)
When the 64-bit result of the multiply-and-accumulate operation that is stored in the multiply-and-accumulate registers MCRH and MCRL is equal to or greater than the maximum positive value for a 32-bit signed numeric value (0x000000007fffffff), the maximum positive value (0x7fffffff) is stored in Dn. If the value stored in the multiply-and-accumulate registers MCRH and MCRL is equal to or less than the maximum negative value for a 32-bit signed numeric value (0xffffffff80000000), the maximum negative value (0x80000000) is stored in Dn. In all other cases, the contents of MCRL are stored in Dn.

(2) When the value of Dm or imm8 is 16 (0x00000010)
When the 64-bit result of the multiply-and-accumulate operation that is stored in the multiply-and-accumulate registers MCRH and MCRL is equal to or greater than the maximum positive value for a 16-bit signed numeric value (0x0000000000007fff), the maximum positive value (0x00007fff) is stored in Dn. If the value stored in the multiply-and-accumulate registers MCRH and MCRL is equal to or less than the maximum negative value for a 16-bit signed numeric value (0xffffffffffff8000), the maximum negative value (0xffff8000) is stored in Dn. In all other cases, the contents of MCRL are stored in Dn.

(3) When the value of Dm or imm8 is 8 (0x00000008)
When the 32-bit result of the multiply-and-accumulate operation that is stored in the multiply-and-accumulate register MCRL is equal to or greater than the maximum positive value for an 8-bit signed numeric value (0x0000007f), the maximum positive value (0x7f) is stored in Dn. If the value stored in the multiply-and-accumulate register MCRL is equal to or less than the maximum negative value for an 8-bit signed numeric value (0xffffff80), the maximum negative value (0x80) is stored in Dn. In all other cases, the contents of MCRL are stored in Dn.

(4) When the value of Dm or imm8 is any other value
The value in Dn is undefined.

[Flag Changes]

When multiply-and-accumulate operation overflow was not detected (MCVF = 0)

Flag	Change	Condition
V	0	Indicates that the multiply-and-accumulate operation is valid.
C	0	Always 0
N	*	Undefined
Z	*	Undefined

When multiply-and-accumulate operation overflow was detected (MCVF = 1)

Flag	Change	Condition
V	1	Indicates that the multiply-and-accumulate operation is invalid.
C	0	Always 0
N	*	Undefined
Z	*	Undefined

[Programming Cautions]

There is a one-instruction delay in the updating of the PSW to reflect flag changes.
However, the Bcc and Lcc instructions can evaluate the flags without waiting for the flag changes to be reflected in the PSW.
The operations of "udf02 imm16, Dn" and "udf02 imm32, Dn" are not assured. In addition, a system error interrupt does not occur in these cases.

MCST9 (Multiply-and-accumulate operation results 9-bit saturation operation instruction/positive value conversion instruction)

[Instruction Format (Macro Name)]

MCST9 Dn

[Assembler Mnemonic]

udf03 Dn, Dn

[Operation]

When the 32-bit result of the multiply-and-accumulate operation that is stored in the multiply-and-accumulate register MCRL is equal to or greater than the maximum positive value for a 9-bit signed numeric value (0x000000ff), the maximum positive value (0xff) is stored in Dn. If the value stored in the multiply-and-accumulate register MCRL is equal to or less than the negative value for a 32-bit signed numeric value (0x00000000), the 0 (0x00) is stored in Dn. In all other cases, the contents of MCRL are stored in Dn.

This instruction sets the contents of the multiply-and-accumulate operation overflow detect register MCVF in the V flag.

[Flag Changes]

When multiply-and-accumulate operation overflow was not detected (MCVF = 0)

Flag	Change	Condition
V	0	Indicates that the multiply-and-accumulate operation is valid.
C	0	Always 0
N	*	Undefined
Z	*	Undefined

When multiply-and-accumulate operation overflow was detected (MCVF = 1)

Flag	Change	Condition
V	1	Indicates that the multiply-and-accumulate operation is invalid.
C	0	Always 0
N	*	Undefined
Z	*	Undefined

[Programming Cautions]

There is a one-instruction delay in the updating of the PSW to reflect flag changes.

However, the Bcc and Lcc instructions can evaluate the flags without waiting for the flag changes to be reflected in the PSW.

When "udf03 Dm, Dn" is operated, Dm is ignored.

The operations of "udf03 imm8, Dn", "udf03 imm16, Dn" and "udf03 imm32, Dn" are not assured. In addition, a system error interrupt does not occur in these cases.

MCST48 (Multiply-and-accumulate operation results 48-bit saturation operation instruction)

[Instruction Format (Macro Name)]

MCST48 Dn

[Assembler Mnemonic]

udf06 Dn, Dn

[Operation]

When the 64-bit result of the multiply-and-accumulate operation that is stored in the multiply-and-accumulate registers MCRH and MCRL is equal to or greater than the maximum positive value for a 48-bit signed numeric value (0x00007fffffffffff), the maximum positive value (0x00007fffffffffff) is output and bits 47 through bits 16 of that output are stored in Dn. If the value stored in the multiply-and-accumulate registers MCRH and MCRL is equal to or less than the maximum negative value for a 48-bit signed numeric value (0xffff800000000000), the maximum negative value (0xffff800000000000) is output and bits 47 through bits 16 of that output are stored in Dn. In all other cases, the contents of MCRH and MCRL are output and bits 47 through bits 16 of that output are stored in Dn.
This instruction sets the contents of the multiply-and-accumulate operation overflow detect register MCVF in the V flag.

[Flag Changes]

When multiply-and-accumulate operation overflow was not detected (MCVF = 0)

Flag	Change	Condition
V	0	Indicates that the multiply-and-accumulate operation is valid.
C	0	Always 0
N	*	Undefined
Z	*	Undefined

When multiply-and-accumulate operation overflow was detected (MCVF = 1)

Flag	Change	Condition
V	1	Indicates that the multiply-and-accumulate operation is invalid.
C	0	Always 0
N	*	Undefined
Z	*	Undefined

[Programming Cautions]

There is a one-instruction delay in the updating of the PSW to reflect flag changes.
However, the Bcc and Lcc instructions can evaluate the flags without waiting for the flag changes to be reflected in the PSW.
When "udf06 Dm, Dn" is operated, Dm is ignored.
The operations of "udf06 imm8, Dn", "udf06 imm16, Dn" and "udf06 imm32, Dn" are not assured. In addition, a system error interrupt does not occur in these cases.

BSCH (Bit search instruction)

[Instruction Format (Macro Name)]

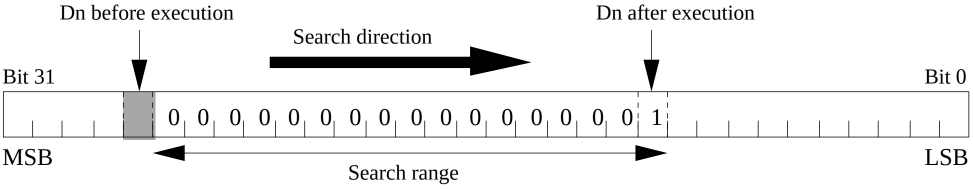
BSCH Dm, Dn

[Assembler Mnemonic]

udf07 Dm, Dn

[Operation]

Bit search is performed within the bit string of the 32 bits contained in Dm from the bit position of the bit number indicated by the contents of Dn - 1 in the direction that the bit number becomes smaller. The bit number of the first bit position where a "1" is found is written into Dn.
When the contents of the lower 5 bits of Dn are 0, bit search is performed from bit 31 in the direction that the bit number becomes smaller.
If search is performed up to the bit position of bit 0 without finding a "1", the C flag is set, Dn is set to 0x00000000, and instruction execution ends.
When instruction execution starts, the upper 27 bits of Dn are ignored.



[Flag Changes]

When search was successful ("1" was found)

Flag	Change	Condition
V	*	Undefined
C	0	This indicates that search was successful.
N	*	Undefined
Z	*	Undefined

When search failed ("1" was not found)

Flag	Change	Condition
V	*	Undefined
C	1	This indicates that search failed.
N	*	Undefined
Z	*	Undefined

[Programming Cautions]

PSW updating by flag changes is delayed by one instruction.
However, Bcc and Lcc instructions can evaluate flags without waiting for flag reflection to PSW.
The operations of "udf07 imm8, Dn", "udf07 imm16, Dn" and "udf07 imm32, Dn" are not assured. In addition, a system error interrupt does not occur in these cases.

SWAP (Data swapping instruction that swaps bytes [high-order to low-order and vice versa] in four-byte data)

[Instruction Format (Macro Name)]

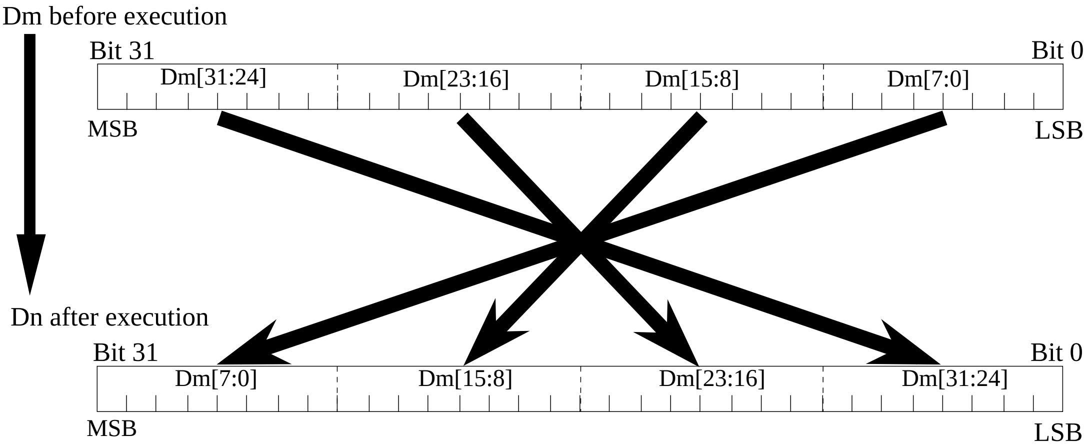
SWAP Dm, Dn

[Assembler Mnemonic]

udf08 Dm, Dn

[Operation]

This instruction swaps the positions of the high-order and low-order 8-bit bytes within the respective high- and low-order 16-bit half-words within the 32-bit data stored in Dm, and then swaps the positions of the high-order and low-order 16-bit half-words, and then stores the result in Dn. As a result, bits 31 through 24 of Dm are stored in bits 7 through 0 in Dn, bits 23 through 16 of Dm are stored in bits 15 through 8 in Dn, bits 15 through 8 of Dm are stored in bits 23 through 16 in Dn, and bits 7 through 0 of Dm are stored in bits 31 through 24 in Dn.



The sample of execution

Before execution:	0x12345678
After execution:	0x78563412

[Flag Changes]

Flag	Change	Condition
V	*	Undefined
C	*	Undefined
N	*	Undefined
Z	*	Undefined

[Programming Cautions]

PSW updating by flag changes is delayed by one instruction.
However, Bcc and Lcc instructions can evaluate flags without waiting for flag reflection to PSW.
The operations of "udf08 imm8, Dn", "udf08 imm16, Dn" and "udf08 imm32, Dn" are not assured. In addition, a system error interrupt does not occur in these cases.

SWAPH (Data swapping instruction [high-order to low-order and vice versa] in two-byte data)

[Instruction Format (Macro Name)]

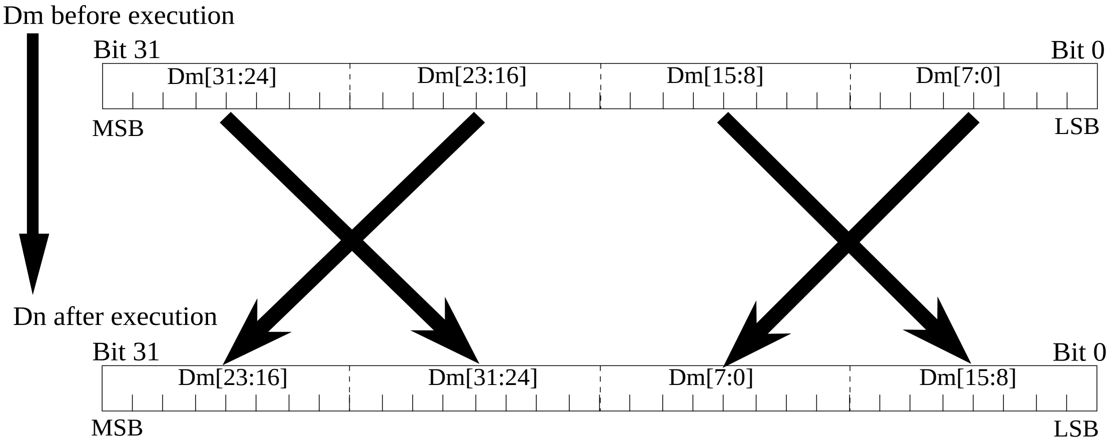
SWAPH Dm, Dn

[Assembler Mnemonic]

udf09 Dm, Dn

[Operation]

This instruction swaps bits 15 through 8 of Dm with bits 7 through 0, and bits 31 through 24 with bits 23 through 16, and then stores the result in Dn.



The sample of execution

Before execution:	0x12345678
After execution:	0x34127856

[Flag Changes]

Flag	Change	Condition
V	*	Undefined
C	*	Undefined
N	*	Undefined
Z	*	Undefined

[Programming Cautions]

PSW updating by flag changes is delayed by one instruction.

However, Bcc and Lcc instructions can evaluate flags without waiting for flag reflection to PSW.

The operations of "udf09 imm8, Dn", "udf09 imm16, Dn" and "udf09 imm32, Dn" are not assured. In addition, a system error interrupt does not occur in these cases.

3.2.4 Programming Notes

■ Notes on instruction description

These programming notes address instruction descriptions as well as instruction placement and combinations. Failure to heed these notes will result in misoperation. A list of these notes is shown below.

Table 3-2-1 Notes on Instruction Description

Preceding instruction	Following instruction	Placement relationship	Notes
Word/half-word data Multiply-and-accumulate instruction *1	Multiply-and-accumulate instruction *3	Following	Insert at least one cycle between the instructions
Word/half-word data Multiply-and-accumulate instruction *1	MCRH, MCRL access instruction *4	Following	Insert at least two cycles between the instructions
Byte data Multiply-and-accumulate instruction *2	MCRH, MCRL access instruction *4	Following	Insert at least one cycle between the instructions
Multiply-and-accumulate instruction *3	Multiply-and-accumulate instruction *3 High-speed multiplication instruction *5	Following	Insert at least three cycles between the instructions
(◆ For details, refer to note (e) on page 3-36.)	Multiply-and-accumulate instruction *3 High-speed multiplication instruction *5	-	Insert at least two NOP instructions immediately before the instructions

*1: The category “Word/half-word data multiply-and-accumulate instruction” applies to the following instructions:
MAC instruction, MACH instruction, MACU instruction, MACHU instruction

*2: The category “byte data multiply-and-accumulate instruction” applies to the following instructions:
MACB instruction, MACBU instruction

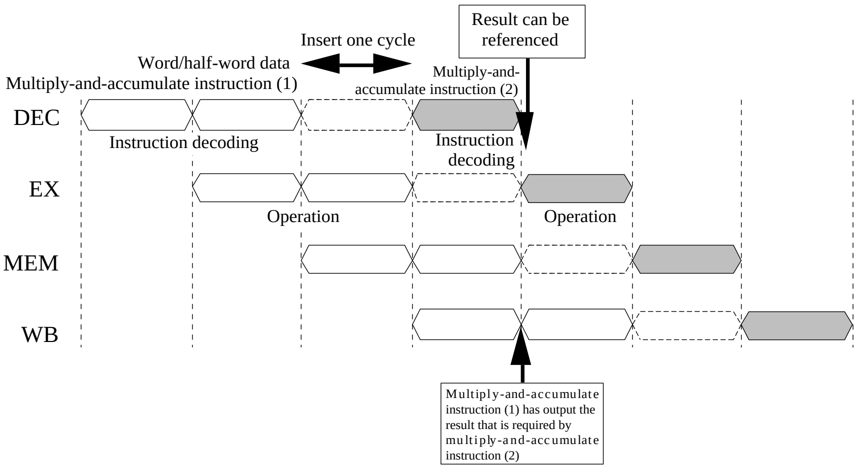
*3: The category “multiply-and-accumulate instruction” applies to the following instructions:
MAC instruction, MACH instruction, MACU instruction, MACHU instruction, MACB instruction, MACBU instruction

*4: The category “MCRH, MCRL access instruction” applies to the following instructions:
PUTCX instruction, CLRMAC instruction, GETCHX instruction, GETCLX instruction

*5: The category “High-speed multiplication instruction” applies to the following instructions:
MULQ instruction, MULQU instruction, MULQI instruction, MULQIU instruction

(a) Note on the description of word/half-word data multiply-and-accumulate instructions and multiply-and-accumulate instructions

When executing a word/half-word data multiply-and-accumulate instruction followed by a multiply-and-accumulate instruction, the result produced by the word/half-word data multiply-and-accumulate instruction is used in the execution of the subsequent multiply-and-accumulate instruction. Therefore, it is essential to not initiate the subsequent multiply-and-accumulate instruction until after the result that is required from the word/half-word data multiply-and-accumulate instruction has been output. As a result, one cycle must be inserted between the word/half-word data multiply-and-accumulate instruction and the subsequent multiply-and-accumulate instruction.


Result can be referenced
Insert one cycle
Word/half-word data
Multiply-and-accumulate instruction (1)
Multiply-and-accumulate instruction (2)
DEC
Instruction decoding
Instruction decoding
EX
Operation
Operation
MEM
WB
Multiply-and-accumulate instruction (1) has output the result that is required by multiply-and-accumulate instruction (2)

This note applies to the following instructions:

<Word/half-word data multiply-and-accumulate instructions>

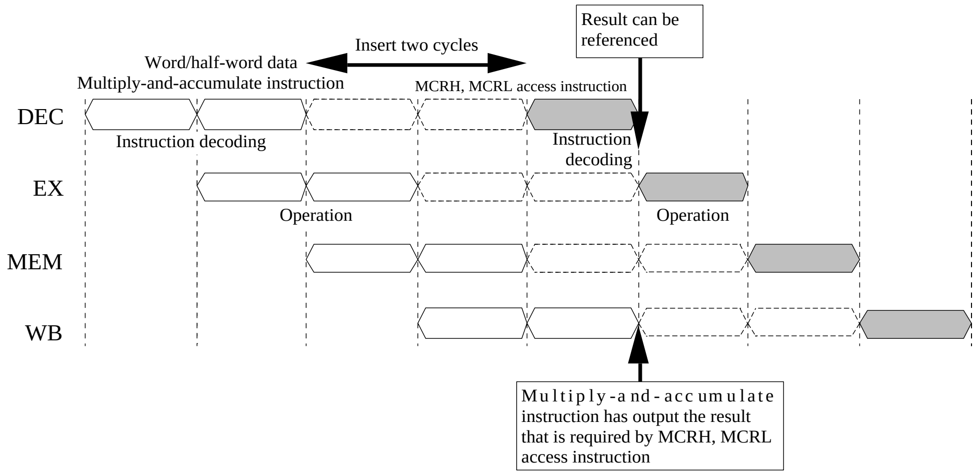
MAC instruction, MACH instruction, MACU instruction, MACHU instruction

<Multiply-and-accumulate instructions>

MAC instruction, MACH instruction, MACU instruction, MACHU instruction, MACB instruction, MACBU instruction

(b) Note on the description of word/half-word data multiply-and-accumulate instructions and MCRH, MCRL access instructions

When executing a word/half-word data multiply-and-accumulate instruction followed by an MCRH, MCRL access instruction, the result produced by the word/half-word data multiply-and-accumulate instruction is used in the execution of the subsequent MCRH, MCRL access instruction. Therefore, it is essential to not initiate the subsequent MCRH, MCRL access instruction until after the result that is required from the word/half-word data multiply-and-accumulate instruction has been output. As a result, two cycles must be inserted between the word/half-word data multiply-and-accumulate instruction and the subsequent MCRH, MCRL access instruction.


Result can be referenced
Insert two cycles
Word/half-word data
Multiply-and-accumulate instruction
MCRH, MCRL access instruction
DEC
Instruction decoding
Instruction decoding
EX
Operation
Operation
MEM
WB
Multiply-and-accumulate instruction has output the result that is required by MCRH, MCRL access instruction

This note applies to the following instructions:

<Word/half-word data multiply-and-accumulate instructions>

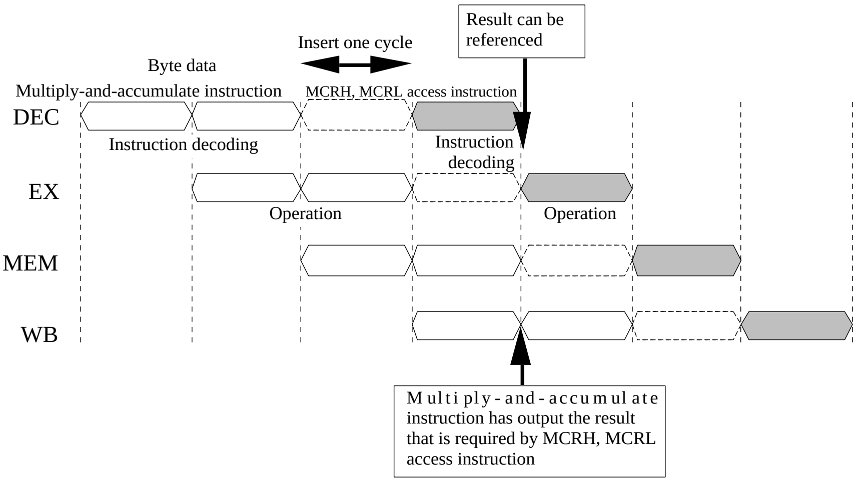
MAC instruction, MACH instruction, MACU instruction, MACHU instruction

<MCRH, MCRL access instructions>

PUTCX instruction, CLRMAC instruction, GETCHX instruction, GETCLX instruction

(c) Note on the description of byte data multiply-and-accumulate instructions and MCRH, MCRL access instructions

When executing a byte data multiply-and-accumulate instruction followed by an MCRH, MCRL access instruction, the result produced by the byte data multiply-and-accumulate instruction is used in the execution of the subsequent MCRH, MCRL access instruction. Therefore, it is essential to not initiate the subsequent MCRH, MCRL access instruction until after the result that is required from the byte data multiply-and-accumulate instruction has been output. As a result, one cycle must be inserted between the byte data multiply-and-accumulate instruction and the subsequent MCRH, MCRL access instruction.


Result can be referenced
Insert one cycle
Byte data
Multiply-and-accumulate instruction
MCRH, MCRL access instruction
DEC
Instruction decoding
Instruction decoding
EX
Operation
Operation
MEM
WB
Multiply-and-accumulate instruction has output the result that is required by MCRH, MCRL access instruction

This note applies to the following instructions:

<Byte data multiply-and-accumulate instructions>

MACB instruction, MACBU instruction

<MCRH, MCRL access instructions>

PUTCX instruction, CLRMAC instruction, GETCHX instruction, GETCLX instruction

(d) Note on the description of multiply-and-accumulate instructions and multiply-and-accumulate instructions or multiply-and-accumulate instructions and quick multiplication instructions

When executing a multiply-and-accumulate instruction followed by another multiply-and-accumulate instruction or a quick multiplication instruction, at least three cycles must be inserted between the instructions. However, no problems are encountered in the case of the instruction combinations listed in the table, or when the value of the multiply-and-accumulate operation overflow detect register MCVF is not used.

Preceding instruction	Following instruction
MAC or MACH	MAC or MACH
MACU or MACHU	MACU or MACHU
MACB	MACB
MACBU	MACBU

This note applies to the following instructions (except in the case of the instruction combinations listed above):

<Multiply-and-accumulate instructions>

MAC instruction, MACH instruction, MACU instruction, MACHU instruction
MACB instruction, MACBU instruction,

<High-speed multiplication instructions>

MULQ instruction, MULQU instruction, MULQI instruction, MULQIU instruction

(e) Note on the description of memory access and multiply-and-accumulate instruction or high-speed multiplication instruction

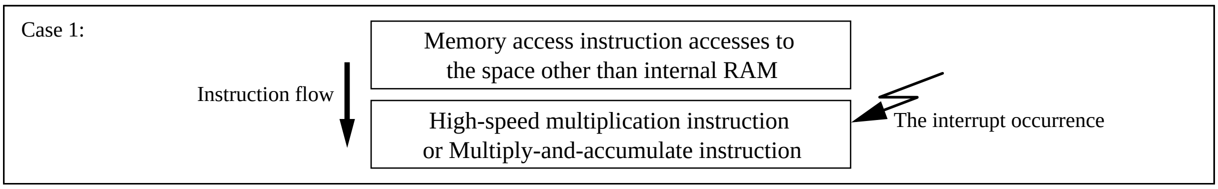
There is an error occasion - CPU hung-up - as written below, if High-speed multiplication instruction or Multiply-and-accumulate instruction is executed within 2 instructions after a memory access instruction that accesses to internal ROM, internal peripheral I/O space or external memory space (this space is referred to as "the space other than internal RAM" below).
However, this note is not applied in either of the following 4 conditions.

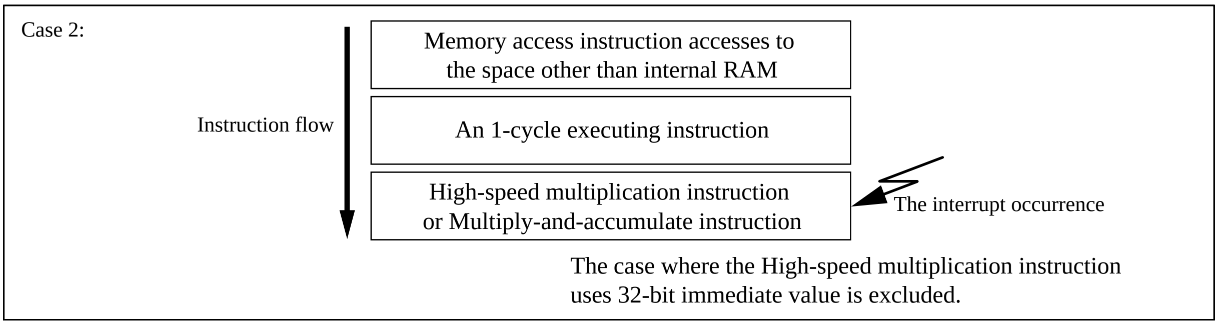
1. The Extension Instruction is not used.
   PanaXSeries C compiler outputs High-speed multiplication instruction only if you use compiler option (-mmulq). If you don't use that option, PanaXSeries C compiler never outputs Extension Instruction.
2. Only High-speed multiplication instructions are used in Extension Instructions.
3. Only Multiply-and-accumulate instructions are used in Extension Instructions.
4. Only the other extension instructions are used in Extension Instructions.

♦ In this note, "Extension Instructions" are classified into Multiply-and-accumulate instructions, High-speed multiplication instructions and the other extension instructions.

There is an error occasion - CPU hung-up -, when "error actualizing condition" occurs after generating "error making potential condition".
"Error making potential condition" and "Error actualizing condition" are described in details below. An "interrupt" on this note is defined as one of level interrupts or non-maskable interrupts*.
* When ICE is used, there is error occasion as in the case of level interrupts and non-maskable interrupts.

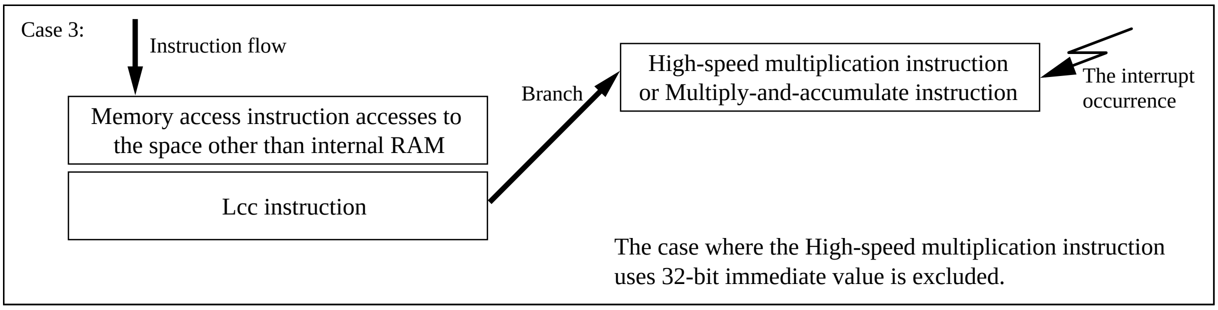
<Error making potential condition>
Error making potential condition occurs when an interrupt is requested during instruction decoding of High-speed multiplication instruction or Multiply-and-accumulate instruction executed after a memory access instruction that accesses to the space other than internal RAM. Error making potential conditions are classified into the following 12 cases.


Case 1:
Memory access instruction accesses to the space other than internal RAM
Instruction flow
High-speed multiplication instruction or Multiply-and-accumulate instruction
The interrupt occurrence


Case 2:
Memory access instruction accesses to the space other than internal RAM
Instruction flow
An 1-cycle executing instruction
High-speed multiplication instruction or Multiply-and-accumulate instruction
The interrupt occurrence

The case where the High-speed multiplication instruction uses 32-bit immediate value is excluded.

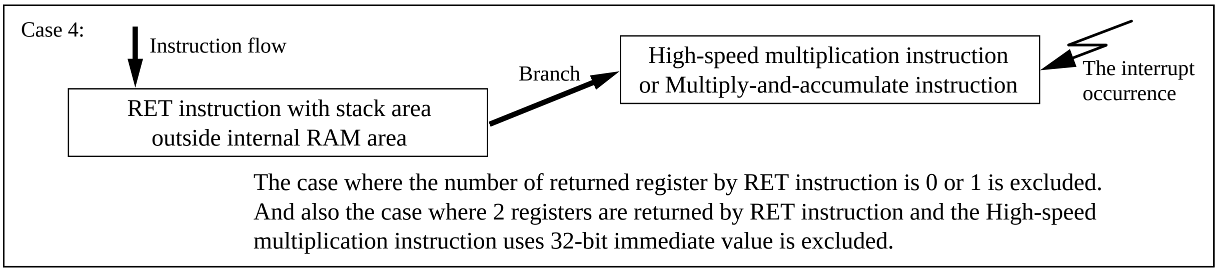
Case 3:


Instruction flow
Memory access instruction accesses to the space other than internal RAM
Lcc instruction
Branch
High-speed multiplication instruction or Multiply-and-accumulate instruction
The interrupt occurrence

The case where the High-speed multiplication instruction uses 32-bit immediate value is excluded.

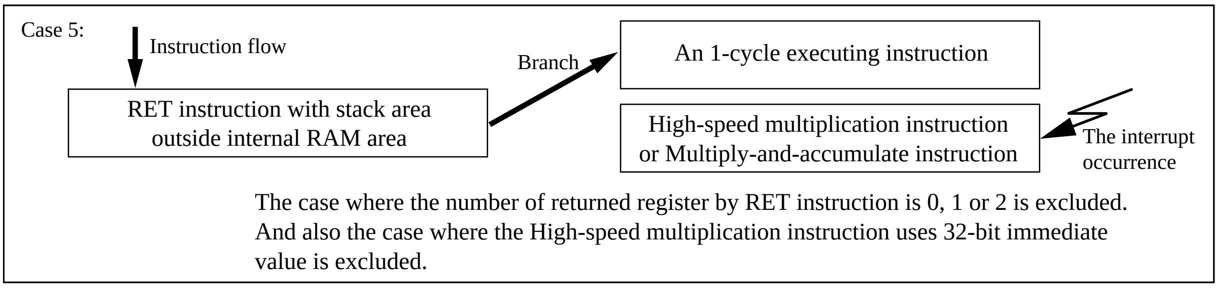
If a stack area is in the internal RAM, any error making potential condition shown on the following cases 4 to 12 is not generated.

Case 4:


Instruction flow
RET instruction with stack area outside internal RAM area
Branch
High-speed multiplication instruction or Multiply-and-accumulate instruction
The interrupt occurrence

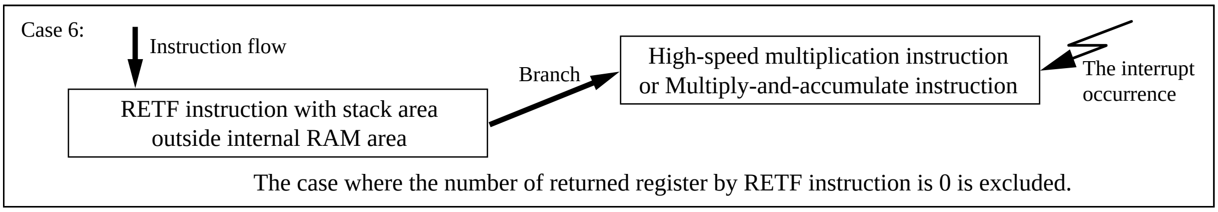
The case where the number of returned register by RET instruction is 0 or 1 is excluded.
And also the case where 2 registers are returned by RET instruction and the High-speed multiplication instruction uses 32-bit immediate value is excluded.

Case 5:


Instruction flow
RET instruction with stack area outside internal RAM area
Branch
An 1-cycle executing instruction
High-speed multiplication instruction or Multiply-and-accumulate instruction
The interrupt occurrence

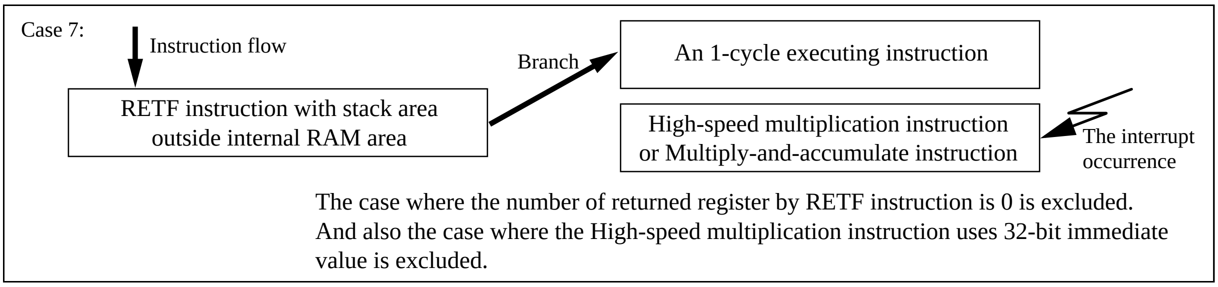
The case where the number of returned register by RET instruction is 0, 1 or 2 is excluded.
And also the case where the High-speed multiplication instruction uses 32-bit immediate value is excluded.

Case 6:


Instruction flow
RETF instruction with stack area outside internal RAM area
Branch
High-speed multiplication instruction or Multiply-and-accumulate instruction
The interrupt occurrence

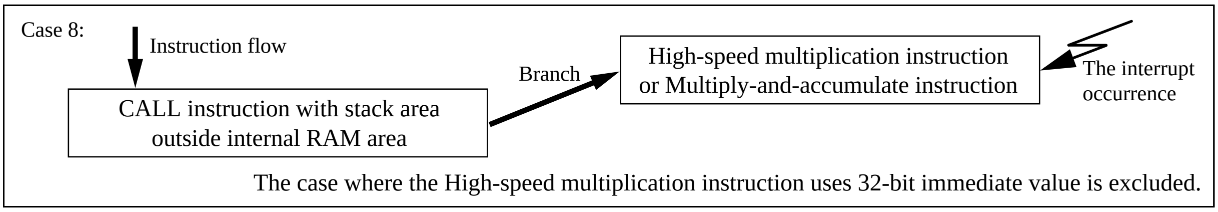
The case where the number of returned register by RETF instruction is 0 is excluded.

Case 7:

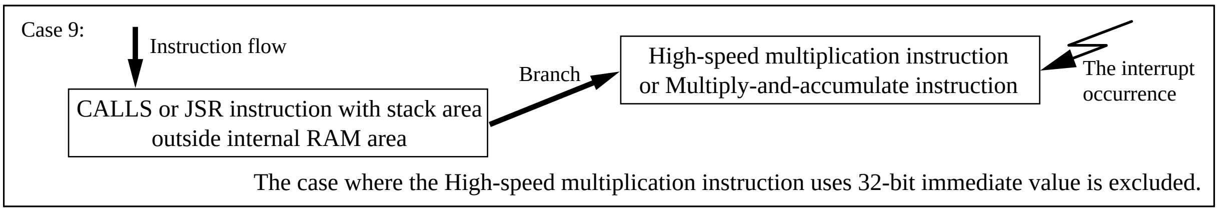

Instruction flow
RETF instruction with stack area outside internal RAM area
Branch
An 1-cycle executing instruction
High-speed multiplication instruction or Multiply-and-accumulate instruction
The interrupt occurrence

The case where the number of returned register by RETF instruction is 0 is excluded.
And also the case where the High-speed multiplication instruction uses 32-bit immediate value is excluded.

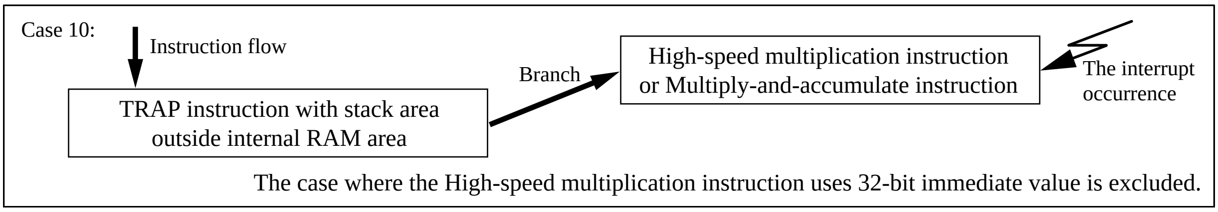
Case 8:


Instruction flow
CALL instruction with stack area outside internal RAM area
Branch
High-speed multiplication instruction or Multiply-and-accumulate instruction
The interrupt occurrence

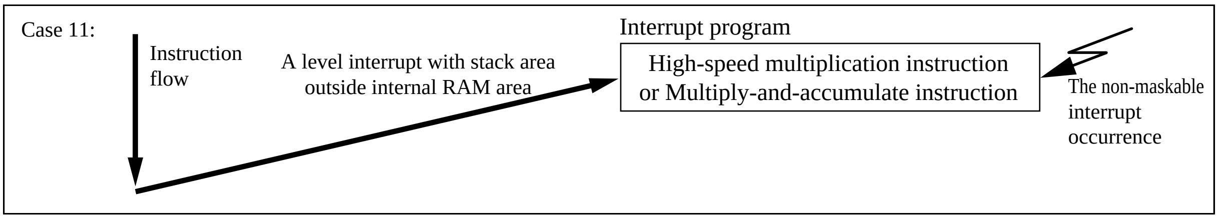
The case where the High-speed multiplication instruction uses 32-bit immediate value is excluded.

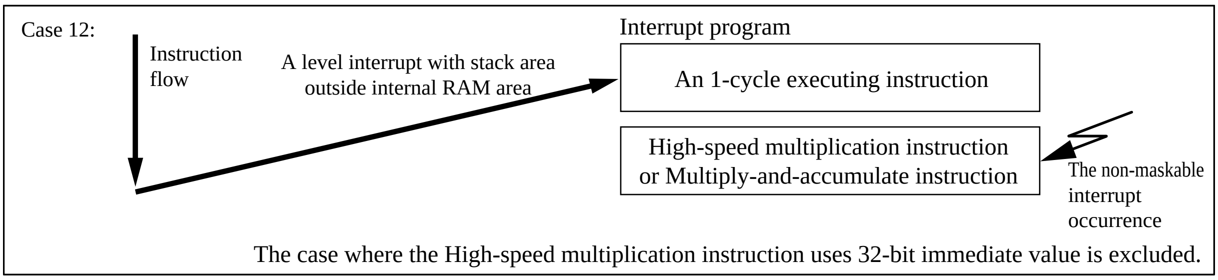

Case 9:
Instruction flow
CALLS or JSR instruction with stack area outside internal RAM area
Branch
High-speed multiplication instruction or Multiply-and-accumulate instruction
The interrupt occurrence

The case where the High-speed multiplication instruction uses 32-bit immediate value is excluded.


Case 10:
Instruction flow
TRAP instruction with stack area outside internal RAM area
Branch
High-speed multiplication instruction or Multiply-and-accumulate instruction
The interrupt occurrence

The case where the High-speed multiplication instruction uses 32-bit immediate value is excluded.


Case 11:
Instruction flow
A level interrupt with stack area outside internal RAM area
Interrupt program
High-speed multiplication instruction or Multiply-and-accumulate instruction
The non-maskable interrupt occurrence


Case 12:
Instruction flow
A level interrupt with stack area outside internal RAM area
Interrupt program
An 1-cycle executing instruction
High-speed multiplication instruction or Multiply-and-accumulate instruction
The non-maskable interrupt occurrence

The case where the High-speed multiplication instruction uses 32-bit immediate value is excluded.

<Error actualizing condition>

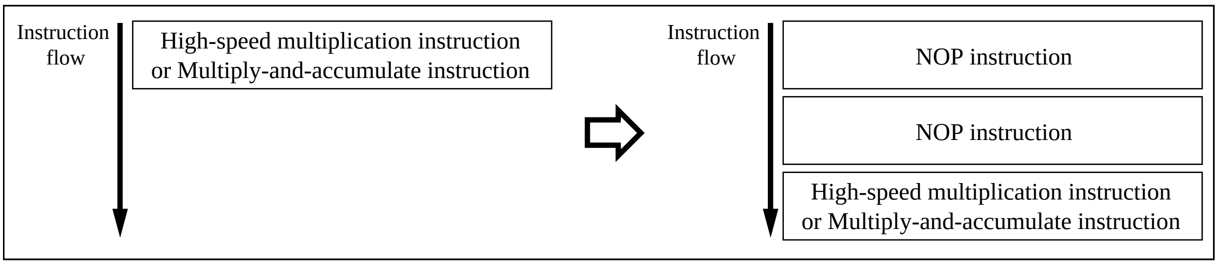
Error actualizing condition is generated by the first extension instruction executed in the interrupt program, or after switching of the task. Error actualizing condition is shown below.

		First extension instruction executed after generating error making potential condition		
		High-speed multiplication instruction	Multiply-and-accumulate instruction	The other extension instruction
Extension instructions which cause to generate error making potential condition	High-speed multiplication instruction	No problem	Error generation	Error generation
	Multiply-and-accumulate instruction	No problem	No problem	Error generation

The error making potential condition is cleared when there is no problem.

When the condition of use corresponds to "Error generation", it is possible to solve the error condition by inserting the 2 NOP instructions as follows.

When the program is compiled and assembled, 2 NOP instructions are inserted on default by using the assembler and linker V3.3R1 or later.


Instruction flow
High-speed multiplication instruction or Multiply-and-accumulate instruction
Instruction flow
NOP instruction
NOP instruction
High-speed multiplication instruction or Multiply-and-accumulate instruction

In addition, please obey the following recommended conditions of 3 points when a program is developed by the assembler so that this error would not occur. As for the program developed by the PanaXSeries C compiler, the following recommended conditions are guaranteed.

1. Please use RTI instruction on a return from an interrupt.
2. Please use the value set by SETLB instruction for LIR and LAR which stores branch target of Lcc instruction.
3. Please don't execute RET instruction or RETF instruction operating the stack frame.

This note applies to the following instructions :

<Memory access instructions>

Ones of the following instructions which access to a memory.
MOV instruction, MOVBU instruction, MOVHU instruction, MOVM instruction, BSET instruction, BCLR instruction, MOVH instruction (Only store) or MOVB instruction (Only store).

<Multiply-and-accumulate instructions>

MAC instruction, MACH instruction, MACU instruction, MACHU instruction, MACB instruction, MACBU instruction

<High-speed multiplication instructions>

MULQ instruction, MULQU instruction, MULQI instruction, MULQIU instruction

<The other extension instructions>

PUTX instruction, PUTCX instruction, GETX instruction, GETCHX instruction, GETCLX instruction, CLRMAC instruction, SAT16 instruction, SAT24 instruction, MCST instruction, MACT9 instruction, MCST48 instruction, BSCH instruction, SWAP instruction, SWAPH instruction

4. Memory Modes

4.1 Memory Mode Types and Selection

This microcontroller has a 32-bit linear address space of up to 4 Gbytes.

The address space is comprised of internal memory space built into the chip and external memory space located outside the chip. The internal memory space can be further divided into internal data space which allows high-speed data access, internal I/O space which contains the I/O ports and control registers built into the chip, and internal instruction space which mainly contains instructions.

Instructions can only be located in the internal instruction space within the internal memory space and in the external memory space. Data can be located in all address spaces, and can be referenced by the MOV instruction. Accordingly, all addressing modes can be used to access data, enabling efficient programming.

The address space differs according to the two memory modes of memory extension mode and processor mode.

For details on the address space in each memory mode, refer to Fig. 4-3-1 and Fig. 4-3-2. When using the register indirect with displacement and register indirect with index addressing modes, make sure that the space (either the internal instruction space, the internal data space, the internal I/O space, or the external memory space) containing the address pointed at by the base registers (Am, An and SP) and the space containing the calculated address are the same.

4.2 Memory Mode Pin Processing

Fix the input levels for the memory mode pins (MMOD0,1) as shown in Table 4-2-1 and Fig. 4-2-1 with pull-up/pull-down resistors.
For details on the pull-up/pull-down resistance, refer to “High-speed Serial Control Card Operation Manual”.

Table 4-2-1 Memory Mode Setting

MMOD1	MMOD0	Memory mode
L	H	Extension memory mode
H	L	Processor mode
L	L	Setting prohibited
H	H	Setting prohibited

For details on the memory mode settings for onboard writing of flash memory in the MN1030F01K, refer to chapter 16, “Internal Flash Memory”.

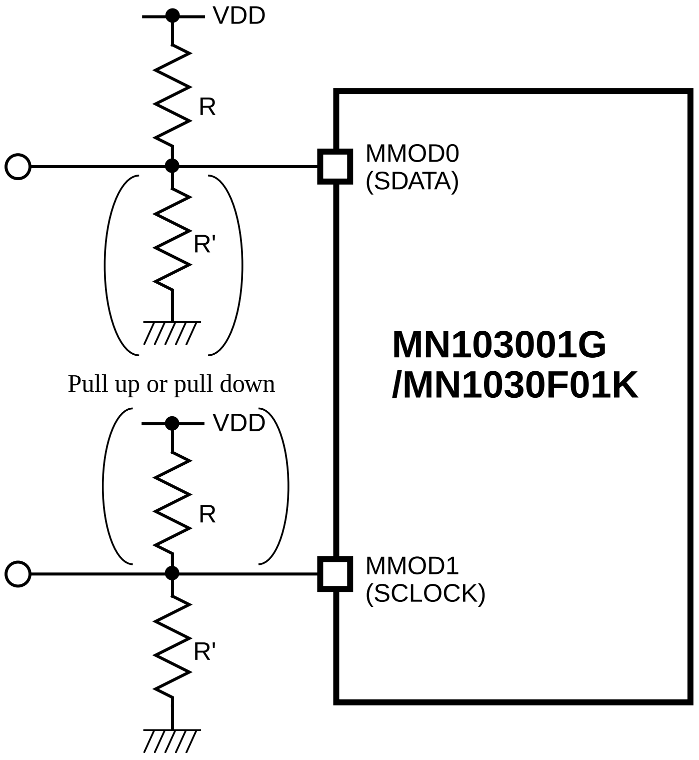


Fig. 4-2-1 Memory Mode Pin Connection Diagram

Note that the memory mode pins (MMOD0,1) also serve as serial interface pins for debugging and for onboard writing of flash memory in the MN1030F01K. The memory mode pins (MMOD0 and 1) are normally input pins, but when they are connected to the serial interface for debugging and for onboard writing of flash memory in the MN1030F01K, they become N ch (when pulled up) or P ch (when pulled down) open drain input/output pins.

♦ Direct inquires for details on the serial interface for debugging and for onboard writing of flash memory in the MN1030F01K to the contact indicated at the end of this manual.

4.3 Description of Memory Mode

4.3.1 Memory Extension Mode

The memory mode which comprises a system from both internal and external memory is called memory extension mode. This mode enables configuration of a system where the program and data make the best use of the high-speed performance of internal memory and the large capacity of external memory. This mode is useful when the program sizes exceed the maximum internal capacity or when locating instructions externally due to facilitate program revisions.

Memory extension mode has memory space of up to 3 GB from addresses x'00000000 to x'BFFFFFFF. Addresses x'00000000 to x'1FFFFFFF are the internal data space (up to 512 MB) which contains data, addresses x'20000000 to x'3FFFFFFF are the internal I/O space (up to 512 MB) which is assigned to the I/O ports and control registers, addresses x'40000000 to x'7FFFFFFF are the internal instruction space which contains instructions and table data, and addresses x'80000000 to x'BFFFFFFF are the external memory space (up to 1 GB).

The MN103001G has 128 Kbytes of internal instruction ROM located at x'40000000 to x'4001FFFF. The MN103001G also has 8 Kbytes of internal data RAM located at x'00000000 to x'00001FFF.

The MN1030F01K has 256 Kbytes of internal flash memory located at x'40000000 to x'4002FFFF. The MN1030F01K also has 8 Kbytes of internal data RAM located at x'00000000 to x'00001FFF.

Note that it is prohibited to access unmounted space of the internal data space, the internal I/O space and the internal instruction space. When accessing the unmounted space, the operation is not assured.

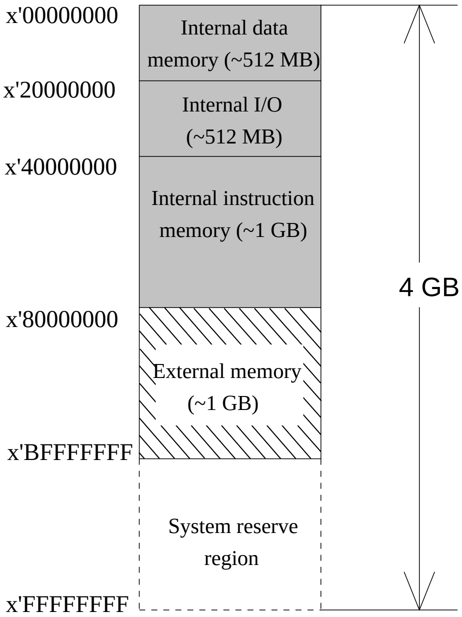


Fig. 4-3-1 Memory Space in Extension Memory Mode

4.3.2 Processor Mode

The memory mode which executes externally located instructions while using the internal data RAM and I/O ports is called processor mode. The internal instruction ROM and the internal flash memory are not used for this mode. Processor mode has memory space of up to 3 GB from addresses x'00000000 to x'BFFFFFFF. Addresses x'00000000 to x'1FFFFFFF are the internal data space (up to 512 MB) which contains data, addresses x'20000000 to x'3FFFFFFF are the internal I/O space (up to 512 MB) which is assigned to the I/O ports and control registers, and addresses x'40000000 to x'BFFFFFFF are the external memory space (up to 2 GB).
This microcontroller has 8 Kbytes of data RAM located at x'00000000 to x'00001FFF.
Note that it is prohibited to access unmounted space of the internal data space and the internal I/O space. When accessing the unmounted space, the operation is not assured.

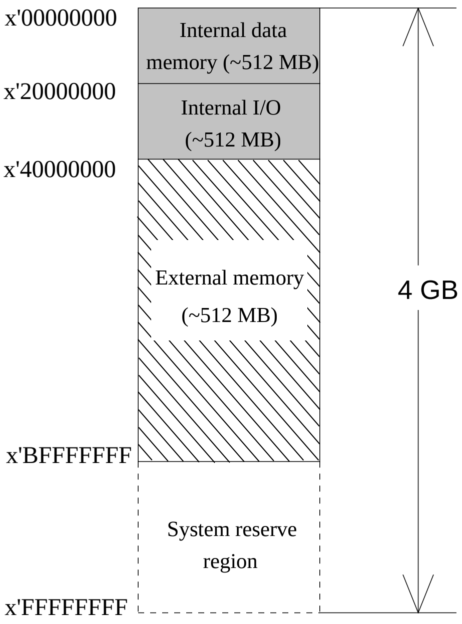


Fig. 4-3-2 Memory Space in Processor Mode

5. Operating Mode

5.1 Overview

The 32-bit microcontroller has the following three operating modes. Oscillator start/stop and CPU and peripheral circuit start/stop switching control functions are provided to support low power consumption.

Operating modes

1. Reset mode (RESET)
2. Normal operation mode (NORMAL)
3. Low power mode
   Stop mode (STOP)
   Halt mode (HALT)
   Sleep mode (SLEEP)

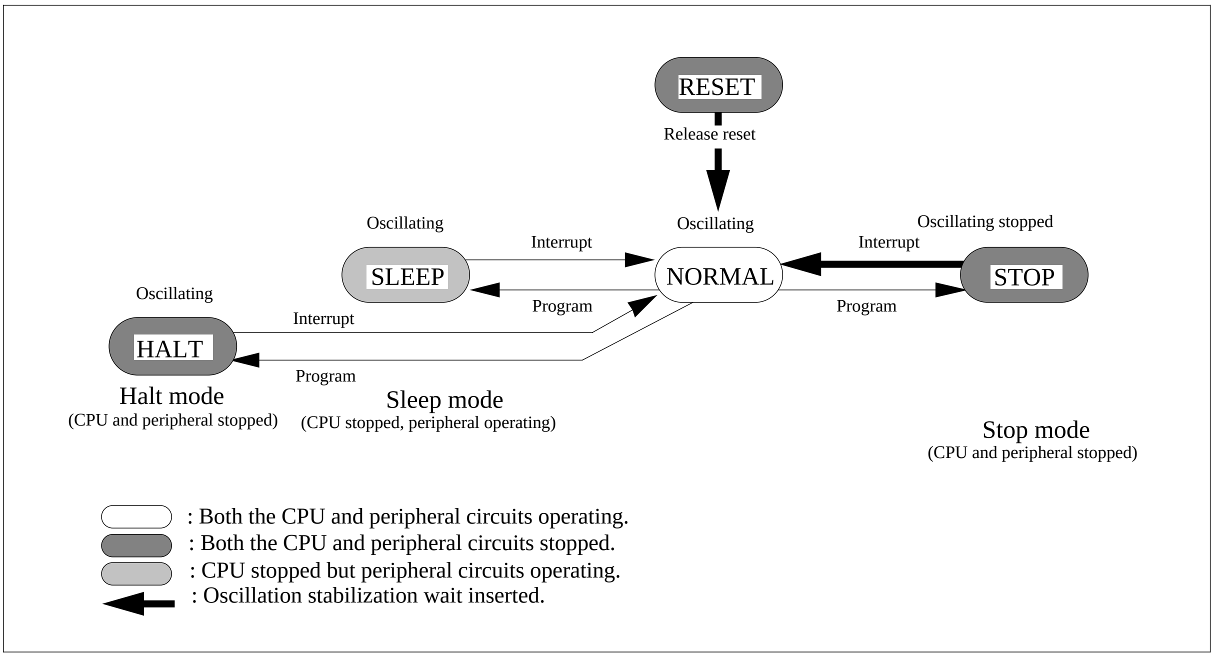

RESET
Release reset
Oscillating
NORMAL
Oscillating
SLEEP
Interrupt
Program
Oscillating
HALT
Interrupt
Program
Oscillating stopped
STOP
Interrupt
Program
Halt mode
(CPU and peripheral stopped)
Sleep mode
(CPU stopped, peripheral operating)
Stop mode
(CPU and peripheral stopped)
: Both the CPU and peripheral circuits operating.
: Both the CPU and peripheral circuits stopped.
: CPU stopped but peripheral circuits operating.
: Oscillation stabilization wait inserted.

Fig. 5-1-1 Operating Mode Transition Diagram

(Notes)

- All modes are entered to normal operating mode by resetting the system.
- Changing the mode by program is performed by setting the CPUM register.

5.2 Reset Mode

- The mode in which the reset ($\overline{\text{RST}}$) pin is active (“L” level) is called “Reset Mode”.
- When the reset pin is low, the chip is reset (initialized) internally. When the reset pin makes the transition to high, the oscillation stabilization wait time is started by an internal 18-bit (when CKSEL pin = “H”) or 19-bit (when CKSEL pin = “L”) binary counter based on the oscillation clock.

The oscillation stabilization wait time t_{OSCW} for oscillation frequency f_{OSC} [MHz] is:
$t_{OSCW} = 2^n / (f_{OSC} \text{ x } 10^3)$ [ms]
n = 18 (when CKSEL pin = “H”) or 19 (when CKSEL pin = “L”)
In other words, when CKSEL pin = “H” and f_{OSC} = 15[MHz]:
t_{OSCW} = 17.476 [ms]

- Table 5-2-1 shows the status of the internal registers immediately after a reset.

Table 5-2-1 Status of Internal Registers Immediately after a Reset

PC	x'40000000
D3 to D0	Undefined
SP	Undefined
A3 to A0	Undefined
MDR	Undefined
LIR	Undefined
LAR	Undefined
PSW	x'0000

- Internal reset is canceled after completion of oscillation stabilization wait time, and the microcontroller changes to the normal operation mode.

5.3 Low Power Mode

Low power consumption is achieved by stopping the oscillation of the oscillators and the clock generator (CG) and stopping the clocks supplied to the CPU and peripheral circuits. Low power mode contains the following three modes and transitions to the three modes are made through software.

Stop mode (STOP)

In this mode, the oscillation of oscillators as well as the CG oscillation are stopped. In the STOP mode, oscillator and CG operation is started by an interrupt and the microcontroller changes to normal operation mode (NORMAL) after waiting for oscillation to be stabilized.

Halt mode (HALT)

In this mode, the oscillators and CG are oscillating but clock supply to the CPU and peripheral circuits is stopped. Thus, CPU and peripheral circuits operation is stopped.
In the HALT mode, the microcontroller changes to normal operation mode (NORMAL) when an interrupt occurs.

Sleep mode (SLEEP)

In this mode, the oscillators and CG are oscillating but clock supply only to the CPU is stopped. Thus, CPU operation is stopped but the peripheral circuits are operating. In SLEEP mode, the microcontroller changes to normal operation mode (NORMAL) when an interrupt occurs.

■ Operation of various peripheral functions in the low power consumption modes

The operation of the peripheral functions in SLEEP, HALT, and STOP mode is shown in the table below.
In SLEEP mode, all peripheral functions operate except for the bus controller and the watchdog timer. In HALT and STOP mode, most peripheral functions are stopped.
In SLEEP mode, the interrupt controller accepts interrupt requests from the peripheral blocks and external pin interrupt requests, notifies the CPU core, and then initiates recovery from SLEEP mode. In HALT/STOP mode, the interrupt controller accepts the external pin interrupt request, notifies the CPU core, and then initiates recovery from HALT/STOP mode.
When making a transition to HALT/SLEEP mode, stop the watchdog timer by clearing the WDCNE bit to "0" in watchdog timer control register WDCTR.

	SLEEP	HALT	STOP
Bus controller	Only responds to DRAM refresh and external bus requests	Stopped	Stopped
Interrupt controller	Operates	Operates	Operates
8-bit timer	Operates	Stopped	Stopped
16-bit timer	Operates	Stopped	Stopped
Watchdog timer	Stopped	Stopped	Stopped
Serial interface	Operates	Stopped	Stopped
A/D converter	Operates	Stopped	Stopped

6. Clock Generator

6.1 Overview

The CG has an internal PLL circuit; in addition to supplying clock pulses to this microcontroller at a frequency that is a multiple of the oscillating frequency of the oscillator, the CG also supplies clock pulses with the same frequency as the oscillating frequency of the oscillator, or that frequency divided by 2, to external devices.

6.2 Features

The features of the CG are described below.

■ Flexible clock control

- Supports self-excitation/external excitation (input frequency: 8 MHz to 20 MHz)
  Note: The in-circuit emulator (ICE) cannot operate with self-excited oscillators in the microcontroller.
- When PLL is being used, a clock that is a programmable multiple of the input frequency is supplied as the CPU clock (MCLK). A clock that is 1/4 of MCLK is supplied as the peripheral clock (IOCLK). A clock that is 1x the input frequency is output as the external device supply clock (SYSCLK).
- When PLL is not being used, a clock that is 1/2 of the input frequency is supplied as the CPU clock (MCLK) and as the external device supply clock (SYSCLK), and a clock that is 1/8 of the input frequency is supplied as the peripheral clock (IOCLK).

6.3 Block Diagram

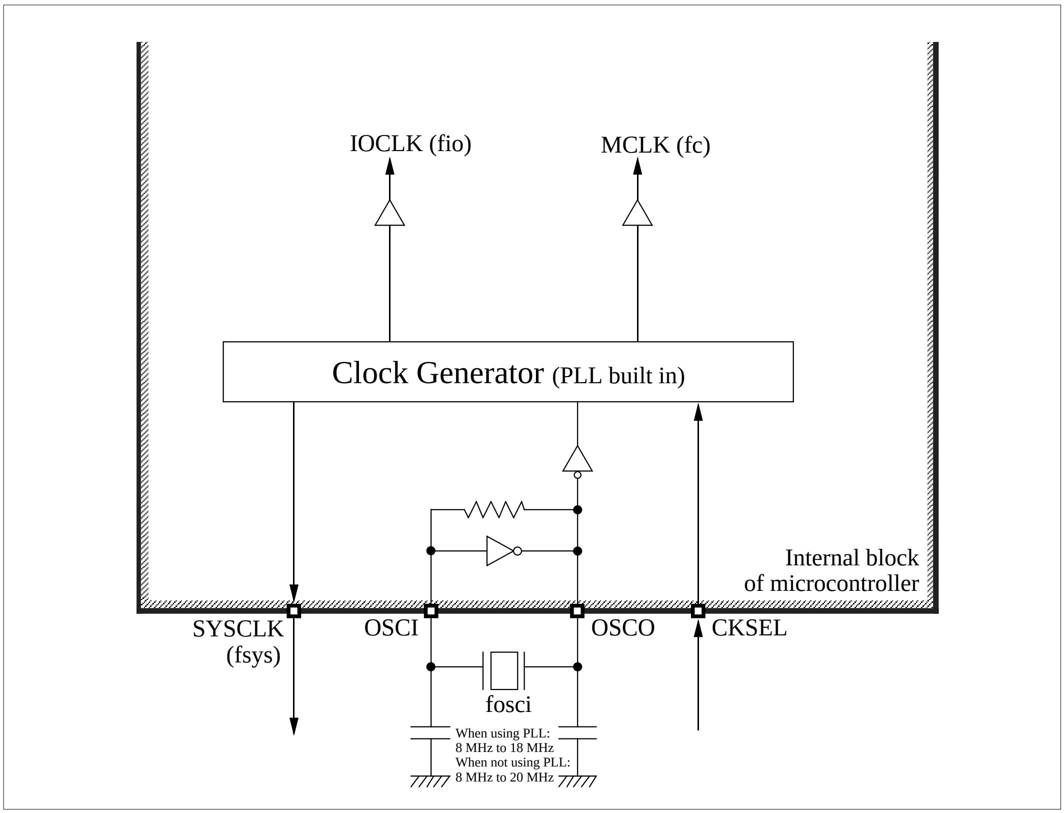


Fig. 6-3-1 Clock Generator

6.4 Description of Operation

6.4.1 Input Frequency Setting

The CG input frequency range is set by the external input pin CKSEL. When CKSEL is set “H”, use an oscillator or resonator with an input frequency fosci such that 8 MHz ≦ fosci ≦ 18 MHz. When CKSEL is set “L”, use an oscillator or resonator with an input frequency fosci such that 8 MHz ≦ fosci ≦ 20 MHz.

Use of an oscillator or resonator that generates a frequency lower than 8 MHz, or higher than 20 MHz is prohibited. The correspondence between the CKSEL mode and the input frequency range is shown in Table 6-4-1.

Table 6-4-1 CKSEL Mode (PLL used/ PLL not used)

Input frequency range	PLL	CKSEL mode
8 MHz ≦ fosci ≦ 18 MHz	When using	H
8 MHz ≦ fosci ≦ 20 MHz	When not using	L

6.4.2 Internal Clock Supply

When external input pin CKSEL is “H”, the frequency of the CPU core/internal RAM/bus controller operation clock (MCLK) is 1x, 2x or 4x the input frequency, depending on the setting of the clock control register, and the frequency of the internal peripheral function operation clock (IOCLK) is 1/4x MCLK. Note that the clock that is supplied to external devices (SYSCLK) has the same frequency as the input frequency.

When external input pin CKSEL is “L”, the frequency of the CPU core/internal RAM/bus controller operation clock (MCLK) is 1/2x the input frequency, and the frequency of the internal peripheral function operation clock (IOCLK) is 1/8x the input frequency. Note that the clock that is supplied to external devices (SYSCLK) 1/2x the input frequency.

Note: For details on the clock control register settings, refer to section 8.6.8, “Clock Control Register.”

When the reset state is released, SYSCLK, MCLK, and IOCLK are supplied starting after a certain oscillation stabilization wait time.

Note: For details on the oscillation stabilization wait time, refer to Chapter 12, “Watchdog Timer.”

Note 1: When a clock is supplied from external, input the clock to the OSCI pin, and leave the OSCO pin open.

Note 2: The in-circuit emulator (ICE) cannot operate with self-excited oscillators in the microcontroller. Use the clock generated in the target system.
When the clock is generated in the target system, supply the clock to the in-circuit emulator main unit through a buffer with adequate drive capability. The in-circuit emulator will not operate correctly if the amplitude of the clock is inadequate, the clock signal is noisy, or the buffer has inadequate drive capability.

The relationship between the input frequency (fosci) and the SYSCLK, MCLK, and IOCLK multiples and frequencies is shown in Table 6-4-2, and the relationship between the input frequency (fosci) and the SYSCLK, MCLK, and IOCLK multiples and frequencies when reset is released is shown in Table 6-4-3.

Table 6-4-2 Relationship between the Oscillation Mode and the SYSCLK, MCLK, and IOCLK Frequencies

Input frequency fosci (MHz)	Oscillation mode		Clock control register setting	SYSCLK		MCLK		IOCLK	
	CKSEL	PLL	Register symbol: CKCTR Address: x'32004000	Multiple	Frequency fsys (MHz)	Multiple	Frequency fc (MHz)	Multiple	Frequency fio (MHz)
8 ≦ fosci ≦ 15	H	Used	MCK [1:0]= 10	1	8 ≦ fsys ≦ 15	4	32 ≦ fc ≦ 60*	1	8 ≦ fio ≦ 15
8 ≦ fosci ≦ 18			MCK [1:0] = 01	1	8 ≦ fsys ≦ 18	2	16 ≦ fc ≦ 36	1/2	4 ≦ fio ≦ 9
			MCK [1:0] = 00			1	8 ≦ fc ≦ 18	1/4	2 ≦ fio ≦ 4.5
8 ≦ fosci ≦ 20	L	Not used	Not used	1/2	4 ≦ fsys ≦ 10	1/2	4 ≦ fc ≦ 10	1/8	1 ≦ fio ≦ 2.5

*: In the case of the MN1030F01K, the maximum frequency for MCLK is 40 MHz.

Table 6-4-3 Relationship between the Input Frequency and the SYSCLK, MCLK, and IOCLK Frequencies When Reset Is Released

Oscillation mode		Clock control register setting	SYSCLK	MCLK	IOCLK
CKSEL	PLL	Register symbol: CKCTR Address: x'32004000	Multiple	Multiple	Multiple
H	Used	MCK [1:0] = 00	1	1	1/4
L	Not used	Not used	1/2	1/2	1/8

Note: When changing the input frequency during operation, be certain to enter the reset mode and set the CKSEL pin to the prescribed value during reset mode.

The input frequency ranges shown here are preliminary. When using this LSI, contact our sales office for the product specifications.

7. Internal Memory

7.1 Overview

The MN103001G has 128 Kbytes of instruction ROM and 8 Kbytes of internal data RAM. The MN1030F01K has 256 Kbytes of flash memory and 8 Kbytes of internal data RAM. The instruction ROM/flash memory and data RAM are connected to the CPU core via a 64-bit bus and a 32-bit bus, respectively.

7.2 Features

The features of the internal memory are listed below.

■ Internal instruction ROM (MN103001G)

- Capacity: 128 Kbytes
- Instruction bus width: 64 bits
- Access cycles: Instruction read: 2 MCLK cycles
  Data read: 3 MCLK cycles

■ Internal flash memory (MN1030F01K)

- Capacity: 256 Kbytes
- Instruction bus width: 64 bits
- Access cycles: Instruction read: 2 MCLK cycles
  Data read: 3 MCLK cycles

■ Internal data RAM (MN103001G/MN1030F01K)

- Capacity: 8 Kbytes
- Data bus width: 32 bits
- Access cycles (*): Data read: 1 MCLK cycle
  Data write: 1 MCLK cycle

(*): The internal data RAM can only be used to store or read data; it is not possible to store or read instructions in the internal data RAM.

7.3 Internal Memory Configuration

The internal instruction ROM is located in the internal memory space at address x'40000000 to x'4001FFFF, while the internal flash memory is located at address x'40000000 to x'4003FFFF and the internal data RAM is located at address x'00000000 to x'00001FFF. Each is connected to the CPU core independently by a dedicated bus in the Harvard architecture (in memory extension mode).

Fig. 7-3-1 shows a block diagram of the internal memory in memory extension mode.

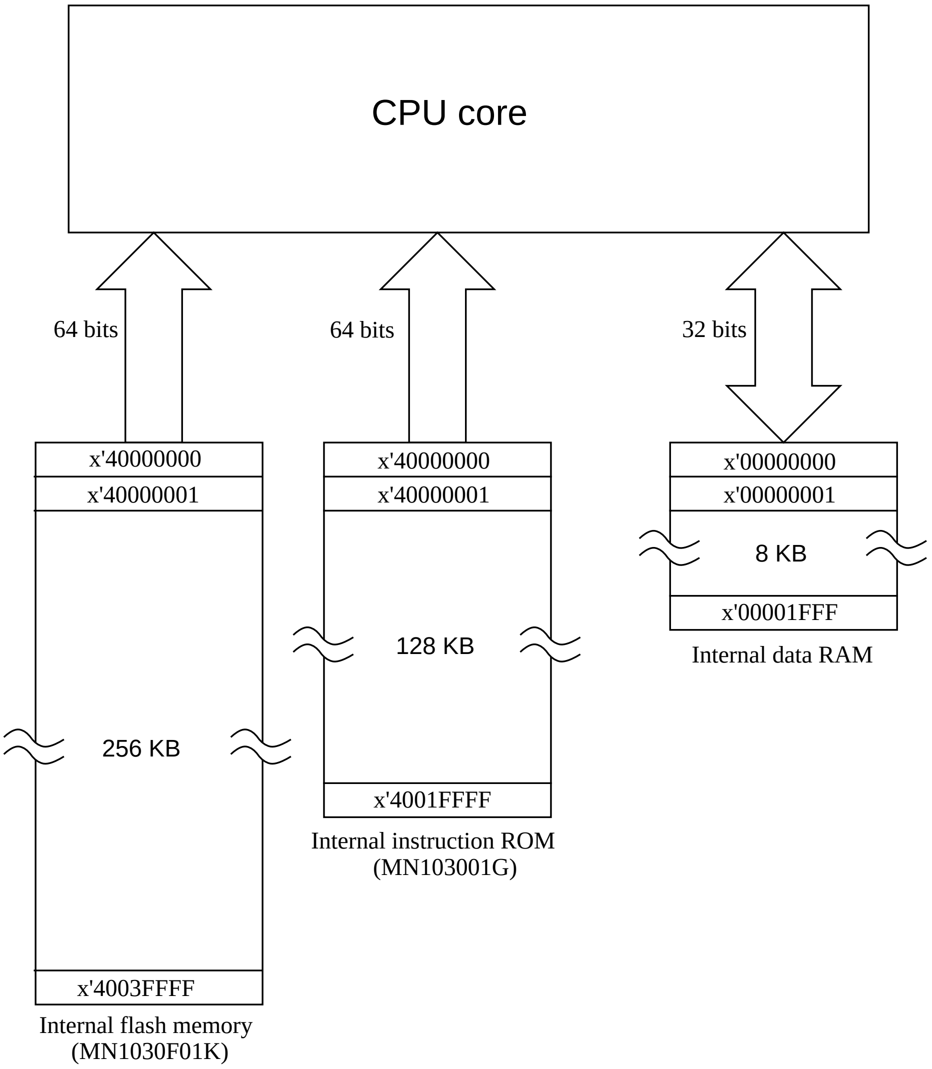


Fig. 7-3-1 Internal Memory Block Diagram (In Memory Extension Mode)

Note: In processor mode, internal instruction ROM and internal flash memory is not connected to the CPU core, and addresses x’40000000 to x’BFFFFFFF are used for external memory.

8. Bus Controller (BC)

8.1 Overview

The bus controller (BC) controls interfacing between the CPU core, internal I/O (peripherals), and devices external to the chip. The bus controller also handles arbitration between the internal and external buses. In addition, in an interface with devices external to the chip, it is possible to select whether address pins and data pins are separate or multiplex. The bus controller outputs four chip select signals, $\overline{\text{RAS}}$/$\overline{\text{CAS}}$ signals, and other signals for an external bus interface, permitting ROM, SRAM, DRAM, and other peripheral LSIs to be connected directly to this microcontroller.

8.2 Features

The features of the bus controller are described below.

■ High-speed control of the internal and external buses through the CPU clock (MCLK) is possible.
- Synchronous mode (synchronized with IOCLK) is supported for the internal I/O bus. Synchronousmode (synchronized with SYSCLK) and asynchronous mode (synchronized with MCLK) are supported for the external bus.

■ External memory space can be partitioned into four blocks
- Chip select signal output for each block
- The bus width can be set to 8 or 16 bits for blocks 0 to 3
- Blocks 0 to 3 can be switched between synchronous mode and asynchronous mode
- Blocks 0 to 3 permit the read/write timing to be set through the software
- Blocks 1 and 2 can be used as DRAM space
- Blocks 2 and 3 permit use for handshaking

■ DRAM interface
- Address multiplexing function
- Permit the read/write timing to be set through the software
- Support for software page mode through software settings
- Support for CAS-before-RAS refresh (Programmable refresh cycle)

■ Permits switching between separate/multiplex address and data pins through the external input pin settings
- Blocks 0 to 3 permit switching between separate/multiplex address and data pins through the external input pin settings
- Using multiplex address and data pins permit the allocation of microcontroller I/O and peripheral pins and reducing the number of external device pins
- Permits direct connection with ROM, SRAM, and DRAM without external circuitry

■ Avoids time penalty during storage operations through use of store buffer (one word)
- Support for storage in on-chip peripheral circuits and external devices
- When the store buffer is empty, storage operations are completed with no wait states, and the CPU can execute subsequent processing

8.3 Bus Configuration

Fig. 8-3-1 shows the bus configuration. The chip's internal buses are the ROM bus between the CPU core and internal instruction ROM/internal flash memory, the RAM bus between the CPU core and internal data RAM, the BC bus between the CPU core and the bus controller, and the I/O bus between the bus controller and internal I/O. The EX bus is an external bus.
Table 8-3-1 lists the characteristics of each bus.

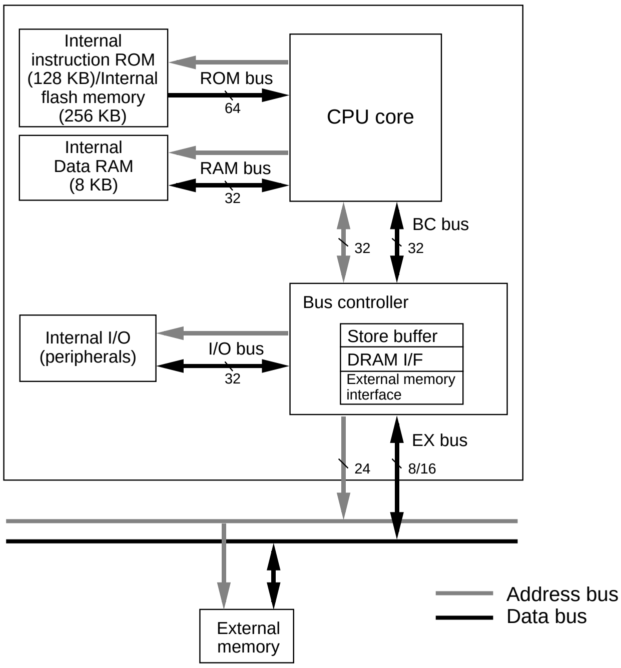


Fig. 8-3-1 Bus Configuration Diagram

Table 8-3-1 Characteristics of Each Bus

Bus name	Blocks	Data bus width	Operating clock
ROM bus	CPU - internal instruction ROM/ internal flash memory	64	MCLK [*1]
RAM bus	CPU - internal data RAM	32	MCLK [*1]
BC bus	CPU - BC	32	MCLK [*1]
I/O bus	BC - internal I/O	32	IOCLK [*1] [synchronous mode]
EX bus (external bus)	BC - external memory	8/16 [*2]	SYSCLK [*1] [synchronous mode] MCLK [*1] [asynchronous mode]

(*1) For a description of the operation clock, refer to section 8.8, "Operation Clock."
(*2) Set by the external input pin or control register.

8.4 Block Diagram

Fig. 8-4-1 shows the block diagram for the bus controller. The bus controller consists of a controller, a store buffer, a CPU interface (BC bus I/F), an interface for internal I/O circuitry (I/O bus I/F) and an external device interface (EX bus I/F).

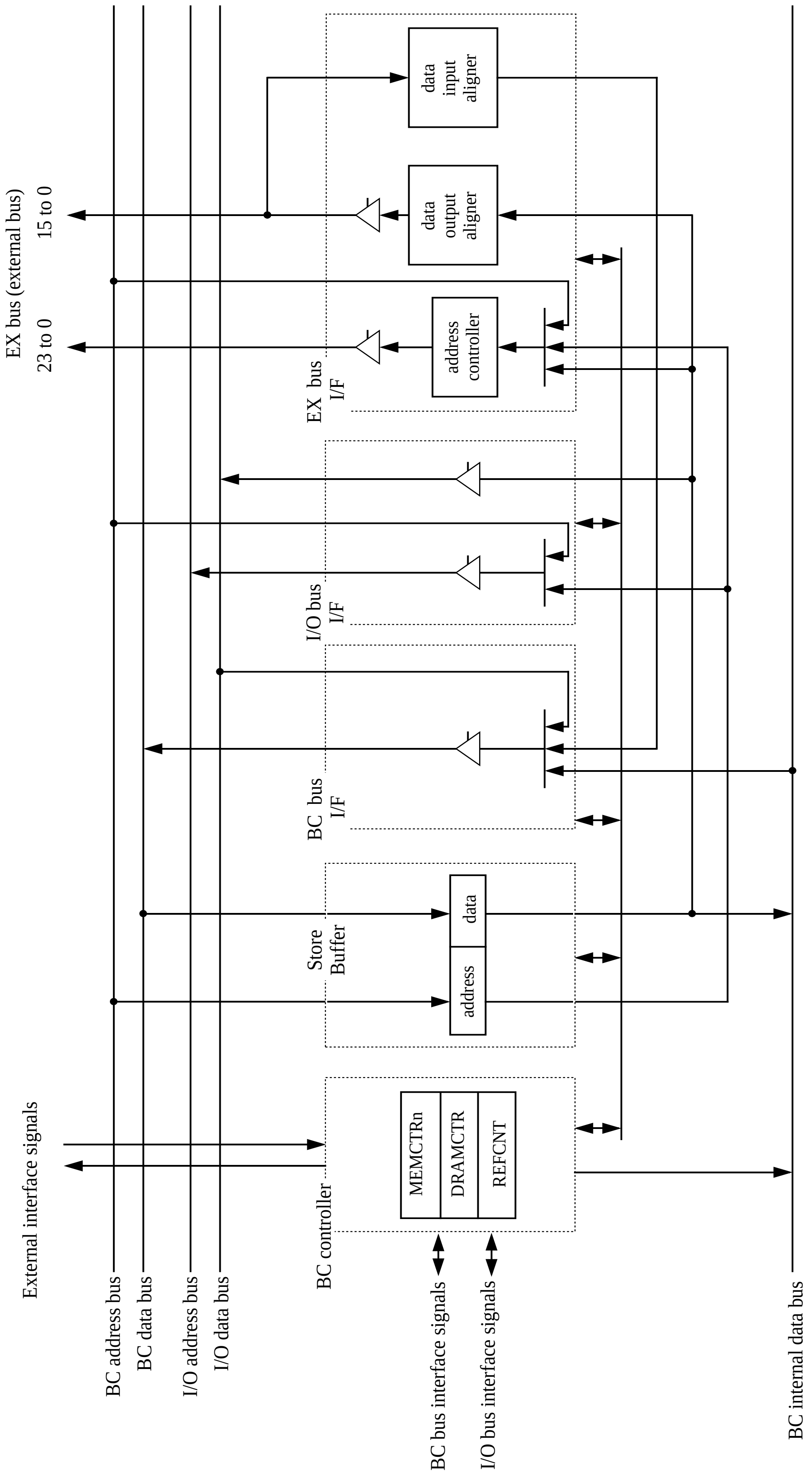


Fig. 8-4-1 Block Diagram for the Bus Controller

8.5 Pin Functions

The external pin functions relating to the bus controller are shown in Table 8-5-1.

Table 8-5-1 External Pin Functions Relating to the Bus Controller

Pin name	Input/output	Number of pins	Function
OSCI	Input	1	Oscillator input pin (when using PLL: 8 MHz to 18 MHz; when not using PLL: 8 MHz to 20 MHz)
OSCO	Output	1	Oscillator output pin (when using PLL: 8 MHz to 18 MHz; when not using PLL: 8 MHz to 20 MHz)
CKSEL	Input	1	Using PLL setting (H: Using; L: Not using)
MMOD1 to 0	Input	2	Memory mode setting signals
EXMOD1 to 0	Input	2	External bus pin mode setting signals
SYSCLK	Output	1	System clock output
A23 to 0	Output	24	Memory address output (Row/column address multiplexed output when DRAM is connected)
D15 to 0	Input/output	16	Memory data input/output
$\overline{\text{RAS 2}}$ to $\overline{\text{1}}$	Output	2	DRAM RAS signals
$\overline{\text{CAS}}$	Output	1	DRAM CAS signal for 2$\overline{\text{WE}}$ ($\overline{\text{WE1}}$ to $\overline{\text{0}}$)
$\overline{\text{DCAS1}}$ to $\overline{\text{0}}$	Output	2	DRAM CAS signals for 2 $\overline{\text{CAS}}$
$\overline{\text{DWE}}$	Output	1	DRAM WE signal for 2 $\overline{\text{CAS}}$
$\overline{\text{CS3}}$ to $\overline{\text{0}}$	Output	4	Chip select signals
$\overline{\text{RE}}$	Output	1	Memory read signal
$\overline{\text{WE1}}$ to $\overline{\text{0}}$	Output	2	Memory write signals (output in byte units)
$\overline{\text{DK}}$	Input	1	Data acknowledge signal
$\overline{\text{BR}}$	Input	1	Bus authority request signal
$\overline{\text{BG}}$	Output	1	Bus authority release signal
ADM15 to 0	Input/output	16	Memory data/memory address input/output (memory address outputs A15 to 0 and memory data D15 to 0 share the same pins)
AS	Output	1	Address strobe signal
RWSEL	Output	1	Read/write select

Note: $\overline{\text{WE}}$1 corresponds to D15 to 8, and $\overline{\text{WE0}}$ corresponds to D7 to 0.
$\overline{\text{CS2}}$ to $\overline{\text{1}}$ and $\overline{\text{RAS2}}$ to $\overline{\text{1}}$, $\overline{\text{CS3}}$ and A23, $\overline{\text{CAS}}$ and A22, AS and D0, RWSEL and D1, and A15 to 0 and ADM15 to 0 are shared pins.

Table 8-5-2 shows the operating status of the external pins concerning BC.

Table 8-5-2 Operating Status of Pins Concerning BC

Operating status Pin	SLEEP mode	STOP mode	HALT mode	When bus is relased
SYSCLK	Operate	L	L	Operate
A23 to 0	Operate	Maintain	Maintain	Hi-Z
ADM15 to 0	Operate	Hi-Z	Hi-Z	Hi-Z
D15 to 0	Operate	Hi-Z	Hi-Z	Hi-Z
$\overline{\text{RAS2}}$ to $\overline{1}$	Operate	H	H	Hi-Z
$\overline{\text{CAS}}$	Operate	H	H	Hi-Z
$\overline{\text{DCAS1}}$ to $\overline{0}$	Operate	H	H	Hi-Z
$\overline{\text{DWE}}$	Operate	H	H	Hi-Z
$\overline{\text{CS3}}$ to $\overline{0}$	Operate	H	H	Hi-Z
$\overline{\text{RE}}$	Operate	H	H	Hi-Z
$\overline{\text{WE1}}$ to $\overline{0}$	Operate	H	H	Hi-Z
$\overline{\text{BG}}$	Operate	H	H	L
AS	Operate	L	L	Hi-Z
RWSEL	Operate	L	L	Hi-Z

Hi-Z:	High impedance
Maintain:	Maintains the status from the previous bus cycle
L:	"L" level output
H:	"H" level output

Note: Because the pins listed in the table at right are all multipurpose, check the status of the external pins of the LSI in chapter 15, “I/O Ports.”
For details on the operating status of each pin upon reset, refer to appendix D, “Pins and Their Operating Statuses upon Reset.”

8.6 Description of Registers

Table 8-6-1 lists the bus controller registers. The settings of these registers are used in timing control, DRAM interface control, etc.

Table 8-6-1 List of Bus Control Registers

Address	Name	Symbol	Number of bits	Initial value	Access size
x'32000020	Memory control register 0B	MEMCTR0B	16	x'EB40	8, 16
x'32000022	Memory control register 1B	MEMCTR1B	16	x'EB50	8, 16
x'32000024	Memory control register 2B	MEMCTR2B	16	x'EB50	8, 16
x'32000026	Memory control register 3B	MEMCTR3B	16	x'EB50	8, 16
x'32000030	Memory control register 0A	MEMCTR0A	16	x'EFFC	8, 16
x'32000032	Memory control register 1A	MEMCTR1A	16	x'EFFC	8, 16
x'32000034	Memory control register 2A	MEMCTR2A	16	x'EFFC	8, 16
x'32000036	Memory control register 3A	MEMCTR3A	16	x'EFFC	8, 16
x'32000040	DRAM control register	DRAMCTR	16	x'0300	8, 16
x'32000042	Refresh count register	REFCNT	16	x'FFFF	8, 16
x'32000044	Page row address register	PRAR	16	x'XXXX	8, 16
x'32004000	Clock control register	CKCTR	16	x'0000	8, 16

8.6.1 Memory Block 0 Control Register

Memory control register 0A/B is used to set the memory block 0 read/write timing and synchronous/asynchronous mode through software.

Memory control register 0A

Register symbol:	MEMCTR0A
Address:	x’32000030
Purpose:	Sets the access timing, etc., for external memory space block 0.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	B0 REN4	B0 REN3	B0 REN2	B0 REN1	B0 REN0	B0 BCE4	B0 BCE3	B0 BCE2	B0 BCE1	B0 BCE0	B0 ADE1	B0 ADE0	B0 EA1	B0 EA0	B0 BCS1	B0 BCS0
Reset	1	1	1	0	1	1	1	1	1	1	1	1	1	1	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: For the external memory access timing charts, refer to section 8.13, “External Memory Space Access (Non-DRAM Spaces).”

Bit No.	Bit name	Description	Setting conditions
1 to 0	BCS1 to 0	Bus cycle start timing When nfr = 2, settings other than “00” or “01” are prohibited. When nfr = 1, settings other than “00” are prohibited.	00: 0MCLK 01: 1MCLK 10: 2MCLK 11: 3MCLK
3 to 2	EA1 to 0	$\overline{\text{RE}}$/$\overline{\text{WE}}$ assert timing	00: 0MCLK ≀ ≀ 11: 3MCLK
5 to 4	ADE1 to 0	Address output end timing	00: 0MCLK ≀ ≀ 11: 3MCLK
10 to 6	BCE4 to 0	Bus cycle end timing Set so that: BCE ≧ REN, BCE ≧ WEN ≧ EA BCE ≧ ASN + ADE	Settings other than those shown below are prohibited. 00100: 4MCLK ≀ ≀ 11111: 31MCLK
15 to 11	REN4 to 0	$\overline{\text{RE}}$ negate timing	Settings other than those shown below are prohibited. 00100: 4MCLK ≀ ≀ 11111: 31MCLK

Note: nfr = MCLK frequency/SYSCLK frequency

Memory control register 0B

Register symbol: MEMCTR0B
Address: x’32000020
Purpose: Sets the bus mode, access timing, etc., for external memory space block 0.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	B0 WEN4	B0 WEN3	B0 WEN2	B0 WEN1	B0 WEN0	B0 ASN2	B0 ASN1	B0 ASN0	B0 ASA1	B0 ASA0	–	–	–	B0 BM	–	–
Reset	1	1	1	0	1	0	1	1	0	1	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R	R

Bit No.	Bit name	Description	Setting conditions
2	BM	Block 0 bus mode	0: Synchronous mode (SYSCLK synchronization) 1: Asynchronous mode (MCLK synchronization)
7 to 6	ASA1 to 0	AS assert timing	00: 0MCLK ≀ ≀ 11: 3MCLK
10 to 8	ASN2 to 0	AS negate timing Set so that: ASN ≧ ASA	000: prohibited 001: 1MCLK ≀ ≀ 111: 7MCLK
15 to 11	WEN4 to 0	$\overline{\text{WE}}$ negate timing Set so that: WEN ≧ EA	Settings other than those shown below are prohibited. 00011: 3MCLK ≀ ≀ 11111: 31MCLK

After the reset is released, block 0 is set as follows:
Synchronous mode

Address output end timing	3MCLK
$\overline{\text{RE}}$ negate timing	29MCLK
$\overline{\text{WE}}$ negate timing	29MCLK
$\overline{\text{RE}}$/$\overline{\text{WE}}$ assert timing	3MCLK
Bus cycle start timing	0MCLK
Bus cycle end timing	31MCLK
AS assert timing	1MCLK
AS negate timing	3MCLK

The bus width is the bus width (8 bits or 16 bits) that accords with the mode specified by the MMOD1 and 0 pins and the EXMOD1 and 0 pins.

Note: For details on the setting of the MMOD1 and 0 pins and the EXMOD1 and 0 pins, refer to section 8.9, "Mode Settings."

8.6.2 Memory Block 1 Control Register

Memory control register 1A/B is used to set the memory block 1 read/write timing, synchronous/asynchronous mode, DRAM mode, page mode, and bus width through software.

Memory control register 1A

Register symbol: MEMCTR1A
Address: x’32000032
Purpose: Sets the access timing, etc., for external memory space block 1.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	B1 REN4	B1 REN3	B1 REN2	B1 REN1	B1 REN0	B1 BCE4	B1 BCE3	B1 BCE2	B1 BCE1	B1 BCE0	B1 ADE1	B1 ADE0	B1 EA1	B1 EA0	B1 BCS1	B1 BCS0
Reset	1	1	1	0	1	1	1	1	1	1	1	1	1	1	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: For the external memory access timing charts, refer to section 8.13, “External Memory Space Access (Non-DRAM Spaces).”
For the timing charts when using DRAM, refer to section 8.14, “External Memory Space Access (DRAM Spaces).”

When not using DRAM (Memory control register 1B B1DRAM = 0)

Bit No.	Bit name	Description	Setting conditions
1 to 0	BCS1 to 0	Bus cycle start timing When nfr = 2, settings other than “00” or “01” are prohibited. When nfr = 1, settings other than “00” are prohibited.	00: 0MCLK 01: 1MCLK 10: 2MCLK 11: 3MCLK
3 to 2	EA1 to 0	$\overline{RE}$/$\overline{WE}$ assert timing	00: 0MCLK ≀ ≀ 11: 3MCLK
5 to 4	ADE1 to 0	Address output end timing	00: 0MCLK ≀ ≀ 11: 3MCLK
10 to 6	BCE4 to 0	Bus cycle end timing Set so that: BCE ≧ REN, BCE ≧ WEN ≧ EA BCE ≧ ASN + ADE	Settings other than those shown below are prohibited. 00100: 4MCLK ≀ ≀ 11111: 31MCLK
15 to 11	REN4 to 0	$\overline{RE}$ negate timing	Settings other than those shown below are prohibited. 00100: 4MCLK ≀ ≀ 11111: 31MCLK

Note: nfr = MCLK frequency/SYSCLK frequency

When using DRAM (Memory control register 1B B1DRAM = 1)

Bit No.	Bit name	Description	Setting conditions
1 to 0	BCS1 to 0	Row address setup timing (use as ASR parameter)	00: prohibited 01: 1MCLK ≀ ≀ 11: 3MCLK
3 to 2	EA1 to 0	Column address setup timing (use as ASC parameter)	00: prohibited 01: 1MCLK ≀ ≀ 11: 3MCLK
5 to 4	ADE1 to 0	Column address output timing (use as CAO parameter) Set so that: CAO (ADE) ≧ ASR (BCS)	00: prohibited 01: 1MCLK ≀ ≀ 11: 3MCLK
8 to 6	BCE2 to 0	$\overline{\text{RAS}}$ hold time (use as RSH parameter)	000: prohibited 001: 1MCLK ≀ ≀ 111: 7MCLK
15 to 11	REN4 to 0	$\overline{\text{CAS}}$ pulse width (use as CAS parameter)	00000: prohibited 00001: 1MCLK ≀ ≀ 11111: 31MCLK

Note: When performing ICE trace/emulation in software page mode, set the CAS parameter to a value of “5” or higher.

Memory control register 1B

Register symbol: MEMCTR1B
Address: x’32000022
Purpose: Sets the bus mode, access timing, etc., for external memory space block 1.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	B1 WEN4	B1 WEN3	B1 WEN2	B1 WEN1	B1 WEN0	B1 ASN2	B1 ASN1	B1 ASN0	B1 ASA1	B1 ASA0	–	B1 BW	B1 PE	B1 BM	–	B1 DRAM
Reset	1	1	1	0	1	0	1	1	0	1	0	1	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W

When not using DRAM (Memory control register 1B B1DRAM = 0)

Bit No.	Bit name	Description	Setting conditions
0	DRAM	Block 1 DRAM space setting	0: Do not use as DRAM space.
2	BM	Block 1 bus mode	0: Synchronous mode (SYSCLK synchronization) 1: Asynchronous mode (MCLK synchronization)
3	PE	Block 1 software page mode enable	Not using
4	BW	Block 1 bus width	0: 8 bits 1: 16 bits
7 to 6	ASA1 to 0	AS assert timing	00: 0MCLK ≀ ≀ 11: 3MCLK
10 to 8	ASN2 to 0	AS negate timing Set so that: ASN ≧ ASA	000: prohibited 001: 1MCLK ≀ ≀ 111: 7MCLK
15 to 11	WEN4 to 0	$\overline{\text{WE}}$ negate timing Set so that: WEN ≧ EA	Settings other than those shown below are prohibited. 00011: 3MCLK ≀ ≀ 11111: 31MCLK

When using DRAM (Memory control register 1B B1DRAM = 1)

Bit No.	Bit name	Description	Setting conditions
0	DRAM	Block 1 DRAM space setting	1: Use as DRAM space.
2	BM	Block 1 bus mode	1: Asynchronous mode (MCLK synchronization)
3	PE	Block 1 software page mode enable	0: Disabled 1: Enabled
4	BW	Block 1 bus width	0: 8 bits 1: 16 bits
7 to 6	ASA1 to 0	Always set to "01".	Any setting other than "01" is prohibited.
10 to 8	ASN2 to 0	$\overline{\text{RAS}}$ precharge cycle (use as RP parameter)	000: prohibited 001: 1MCLK ≀ ≀ 111: 7MCLK
15 to 11	WEN4 to 0	$\overline{\text{WE}}$ negate timing Set so that: CAO (ADE)+CAS (REN) ≧WEN	Settings other than those shown below are prohibited. 00100: 4MCLK ≀ ≀ 11111: 31MCLK

After the reset is released, block 1 is set as follows:

Address output end timing	3MCLK
$\overline{\text{RE}}$ negate timing	29MCLK
$\overline{\text{WE}}$ negate timing	29MCLK
$\overline{\text{RE}}$/$\overline{\text{WE}}$ assert timing	3MCLK
Bus cycle start timing	0MCLK
Bus cycle end timing	31MCLK
AS assert timing	1MCLK
AS negate timing	3MCLK

The bus width is 16 bits, and synchronous mode is set.

8.6.3 Memory Block 2 Control Register

Memory control register 2A/B is used to set the memory block 2 read/write timing, synchronous/asynchronous mode, fixed wait/handshaking mode, DRAM mode, page mode, and bus width through software.

Memory control register 2A

Register symbol: MEMCTR2A
Address: x’32000034
Purpose: Sets the access timing, etc., for external memory space block 2.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	B2 REN4	B2 REN3	B2 REN2	B2 REN1	B2 REN0	B2 BCE4	B2 BCE3	B2 BCE2	B2 BCE1	B2 BCE0	B2 ADE1	B2 ADE0	B2 EA1	B2 EA0	B2 BCS1	B2 BCS0
Reset	1	1	1	0	1	1	1	1	1	1	1	1	1	1	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: For the external memory access timing charts, refer to section 8.13, “External Memory Space Access (Non-DRAM Spaces).”
For the timing charts when using DRAM, refer to section 8.14, “External Memory Space Access (DRAM Spaces).”

When using fixed wait mode and not using DRAM (Memory control register 2B B2DRAM = 0, B2WM = 0)

Bit No.	Bit name	Description	Setting conditions
1 to 0	BCS1 to 0	Bus cycle start timing When nfr = 2, settings other than “00” or “01” are prohibited. When nfr = 1, settings other than “00” are prohibited.	00: 0MCLK 01: 1MCLK 10: 2MCLK 11: 3MCLK
3 to 2	EA1 to 0	$\overline{\text{RE}}$/$\overline{\text{WE}}$ assert timing	00: 0MCLK ≀ ≀ 11: 3MCLK
5 to 4	ADE1 to 0	Address output end timing	00: 0MCLK ≀ ≀ 11: 3MCLK
10 to 6	BCE4 to 0	Bus cycle end timing Set so that: BCE ≧ REN, BCE ≧ WEN ≧ EA BCE ≧ASN + ADE	Settings other than those shown below are prohibited. 00100: 4MCLK ≀ ≀ 11111: 31MCLK
15 to 11	REN4 to 0	$\overline{\text{RE}}$ negate timing	Settings other than those shown below are prohibited. 00100: 4MCLK ≀ ≀ 11111: 31MCLK

Note: nfr = MCLK frequency/SYSCLK frequency

When using handshaking mode (Memory control register 2B B2DRAM = 0, B2WM = 1)

Bit No.	Bit name	Description	Setting conditions
1 to 0	BCS1 to 0	$\overline{\text{DK}}$ detection wait cycle (use as DW parameter)	00: prohibited 01: 1MCLK 10: 2MCLK 11: 3MCLK
3 to 2	EA1 to 0	$\overline{\text{RE}}$/$\overline{\text{WE}}$ assert timing	00: prohibited 01: 1MCLK 10: 2MCLK 11: 3MCLK
5 to 4	ADE1 to 0	Address output end timing	00: 0MCLK ≀ ≀ 11: 3MCLK
10 to 6	BCE4 to 0	Bus cycle end timing Set so that: BCE ≧ REN, BCE ≧ WEN	00000: 0MCLK ≀ ≀ 11111: 31MCLK
15 to 11	REN4 to 0	$\overline{\text{RE}}$ negate timing	00000: 0MCLK ≀ ≀ 11111: 31MCLK

When using DRAM (Memory control register 2B B2DRAM = 1 B2WM = 0)

Bit No.	Bit name	Description	Setting conditions
1 to 0	BCS1 to 0	Row address setup timing (use as ASR parameter)	00: prohibited 01: 1MCLK ≀ ≀ 11: 3MCLK
3 to 2	EA1 to 0	Column address setup timing (use as ASC parameter)	00: prohibited 01: 1MCLK ≀ ≀ 11: 3MCLK
5 to 4	ADE1 to 0	Column address output timing (use as CAO parameter) Set so that: CAO (ADE) ≧ ASR (BCS)	00: prohibited 01: 1MCLK ≀ ≀ 11: 3MCLK
8 to 6	BCE2 to 0	$\overline{\text{RAS}}$ hold time (use as RSH parameter)	000: prohibited 001: 1MCLK ≀ ≀ 111: 7MCLK
15 to 11	REN4 to 0	$\overline{\text{CAS}}$ pulse width (use as CAS parameter)	00000: prohibited 00001: 1MCLK ≀ ≀ 11111: 31MCLK

Note: When performing ICE trace/emulation in software page mode, set the CAS parameter to a value of “5” or higher.

Memory control register 2B

Register symbol: MEMCTR2B
Address: x’32000024
Purpose: Sets the bus mode, access timing, etc., for external memory space block 2.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	B2 WEN4	B2 WEN3	B2 WEN2	B2 WEN1	B2 WEN0	B2 ASN2	B2 ASN1	B2 ASN0	B2 ASA1	B2 ASA0	–	B2 BW	B2 PE	B2 BM	B2 WM	B2 DRAM
Reset	1	1	1	0	1	0	1	1	0	1	0	1	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

When using fixed wait mode and not using DRAM
(Memory control register 2B B2DRAM = 0, B2WM = 0)

Bit No.	Bit name	Description	Setting conditions
0	DRAM	Block 2 DRAM space setting	0: Do not use as DRAM space.
1	WM	Block 2 wait mode	0: fixed wait mode
2	BM	Block 2 bus mode	0: Synchronous mode (SYSCLK synchronization) 1: Asynchronous mode (MCLK synchronization)
3	PE	Block 2 software page mode enable	Not using
4	BW	Block 2 bus width	0: 8 bits 1: 16 bits
7 to 6	ASA1 to 0	AS assert timing	00: 0MCLK ≀ ≀ 11: 3MCLK
10 to 8	ASN2 to 0	AS negate timing Set so that: ASN ≧ ASA	000: prohibited 001: 1MCLK ≀ ≀ 111: 7MCLK
15 to 11	WEN4 to 0	$\overline{\text{WE}}$ negate timing Set so that: WEN ≧ EA	Settings other than those shown below are prohibited. 00011: 3MCLK ≀ ≀ 11111: 31MCLK

When using handshaking mode (Memory control register 2B B2DRAM = 0, B2WM = 1)

Bit No.	Bit name	Description	Setting conditions
0	DRAM	Block 2 DRAM space setting	0: Do not use as DRAM space.
1	WM	Block 2 wait mode	1: Handshaking mode
2	BM	Block 2 bus mode	0: Synchronous mode (SYSCLK synchronization)
3	PE	Block 2 software page mode enable	Not using
4	BW	Block 2 bus width	0: 8 bits 1: 16 bits
7 to 6	ASA1 to 0	AS assert timing	00: 0MCLK ≀ ≀ 11: 3MCLK
10 to 8	ASN2 to 0	AS negate timing Set so that: ASN ≧ ASA	000: prohibited 001: 1MCLK ≀ ≀ 111: 7MCLK
15 to 11	WEN4 to 0	$\overline{\text{WE}}$ negate timing	00000: 0MCLK ≀ ≀ 11111: 31MCLK

When using DRAM (Memory control register 2B B2DRAM = 1 B2WM = 0)

Bit No.	Bit name	Description	Setting conditions
0	DRAM	Block 2 DRAM space setting	1: Use as DRAM space
1	WM	Block 2 wait mode	0: fixed wait mode
2	BM	Block 2 bus mode	1: Asynchronous mode (MCLK synchronization)
3	PE	Block 2 software page mode enable	0: Disable 1: Enable
4	BW	Block 2 bus width	0: 8 bits 1: 16 bits
7 to 6	ASA1 to 0	Always set 01	Settings other than 01 are prohibited.
10 to 8	ASN2 to 0	$\overline{\text{RAS}}$ precharge cycle Use as parameter RP	000: prohibited 001: 1MCLK ≀ ≀ 111: 7MCLK
15 to 11	WEN4 to 0	$\overline{\text{WE}}$ negate timing Set so that: CAO (ADE) + CAS (REN) ≧ WEN	Settings other than those shown below are prohibited. 00100: 4MCLK ≀ ≀ 11111: 31MCLK

After the reset is released, block 2 is set as follows:

Address output end timing	3MCLK
$\overline{\text{RE}}$ negate timing	29MCLK
$\overline{\text{WE}}$ negate timing	29MCLK
$\overline{\text{RE}}$/$\overline{\text{WE}}$ assert timing	3MCLK
Bus cycle start timing	0MCLK
Bus cycle end timing	31MCLK
AS assert timing	1MCLK
AS negate timing	3MCLK

The bus width is 16 bits, and synchronous fixed wait mode is set.

8.6.4 Memory Block 3 Control Register

Memory control register 3A/B is used to set the memory block 3 read/write timing, synchronous/asynchronous mode, fixed wait/handshaking mode, and bus width through software. However, the handshaking mode can only be set when (MCLK frequency/SYSCLK frequency) = 4.

Memory control register 3A

Register symbol: MEMCTR3A
Address: x'32000036
Purpose: Sets the access timing, etc., for external memory space block 3.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	B3 REN4	B3 REN3	B3 REN2	B3 REN1	B3 REN0	B3 BCE4	B3 BCE3	B3 BCE2	B3 BCE1	B3 BCE0	B3 ADE1	B3 ADE0	B3 EA1	B3 EA0	B3 BCS1	B3 BCS0
Reset	1	1	1	0	1	1	1	1	1	1	1	1	1	1	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: For the external memory access timing charts, refer to section 8.13, "External Memory Space Access (Non-DRAM Spaces)."

When using fixed wait mode (Memory control register 3B B3WM = 0)

Bit No.	Bit name	Description	Setting conditions
1 to 0	BCS1 to 0	Bus cycle start timing When nfr = 2, settings other than "00" or "01" are prohibited. When nfr = 1, settings other than "00" are prohibited.	00: 0MCLK 01: 1MCLK 10: 2MCLK 11: 3MCLK
3 to 2	EA1 to 0	$\overline{RE}$/$\overline{WE}$ assert timing	00: 0MCLK ≀ ≀ 11: 3MCLK
5 to 4	ADE1 to 0	Address output end timing	00: 0MCLK ≀ ≀ 11: 3MCLK
10 to 6	BCE4 to 0	Bus cycle end timing Set so that: BCE ≧ REN, BCE ≧ WEN ≧ EA BCE ≧ASN + ADE	Settings other than those shown below are prohibited. 00100: 4MCLK ≀ ≀ 11111: 31MCLK
15 to 11	REN4 to 0	$\overline{RE}$ negate timing	Settings other than those shown below are prohibited. 00100: 4MCLK ≀ ≀ 11111: 31MCLK

Note: nfr = MCLK frequency/SYSCLK frequency

Note: When fixed wait asynchronous mode (B3WM = 0 and B3BM = 1 in MEMCTR3B) is set, "00" must be set for BCS1 to 0.

When using handshaking mode (Memory control register 3B B3WM = 1)

Bit No.	Bit name	Description	Setting conditions
1 to 0	BCS1 to 0	$\overline{\text{DK}}$ detection wait cycle (used as parameter DW)	00: prohibited 01: 1MCLK 10: 2MCLK 11: 3MCLK
3 to 2	EA1 to 0	$\overline{\text{RE}}$/$\overline{\text{WE}}$ assert timing	00: prohibited 01: 1MCLK 10: 2MCLK 11: 3MCLK
5 to 4	ADE1 to 0	Address output end timing	00: 0MCLK ≀ ≀ 11: 3MCLK
10 to 6	BCE4 to 0	Bus cycle end timing Set so that: BCE ≧ REN, BCE ≧ WEN	00000: 0MCLK ≀ ≀ 11111: 31MCLK
15 to 11	REN4 to 0	$\overline{\text{RE}}$ negate timing	00000: 0MCLK ≀ ≀ 11111: 31MCLK

Note: Handshaking mode can only be set when (MCLK frequency/SYSCLK frequency) = 4.
If (MCLK frequency/SYSCLK frequency) = 1 or 2, set B3WM = 0 in MEMCTR3B.

Memory control register 3B

Register symbol: MEMCTR3B
Address: x’32000026
Purpose: Sets the bus mode, access timing, etc., for external memory space block 3.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	B3 WEN4	B3 WEN3	B3 WEN2	B3 WEN1	B3 WEN0	B3 ASN2	B3 ASN1	B3 ASN0	B3 ASA1	B3 ASA0	–	B3 BW	–	B3 BM	B3 WM	–
Reset	1	1	1	0	1	0	1	1	0	1	0	1	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R	R/W	R/W	R

When using fixed wait mode (Memory control register 3B B3WM = 0)

Bit No.	Bit name	Description	Setting conditions
1	WM	Block 3 wait mode	0: fixed wait mode
2	BM	Block 3 bus mode	0: Synchronous mode (SYSCLK synchronization) 1: Asynchronous mode (MCLK synchronization)
4	BW	Block 3 bus width	0: 8 bits 1: 16 bits
7 to 6	ASA1 to 0	AS assert timing	00: 0MCLK ≀ ≀ 11: 3MCLK
10 to 8	ASN2 to 0	AS negate timing Set so that: ASN ≧ ASA	000: prohibited 001: 1MCLK ≀ ≀ 111: 7MCLK
15 to 11	WEN4 to 0	$\overline{\text{WE}}$ negate timing Set so that: WEN ≧ EA	Settings other than those shown below are prohibited. 00011: 3MCLK ≀ ≀ 11111: 31MCLK

When using handshaking mode (Memory control register 3B B3WM = 1)

Bit No.	Bit name	Description	Setting conditions
1	WM	Block 3 wait mode	1: Handshaking mode
2	BM	Block 3 bus mode	0: Synchronous mode (SYSCLK synchronization)
4	BW	Block 3 bus width	0: 8 bits 1: 16 bits
7 to 6	ASA1 to 0	AS assert timing	00: 0MCLK ≀ ≀ 11: 3MCLK
10 to 8	ASN2 to 0	AS negate timing Set so that: ASN ≧ ASA	000: prohibited 001: 1MCLK ≀ ≀ 111: 7MCLK
15 to 11	WEN4 to 0	$\overline{\mathrm{WE}}$ negate timing	00000: 0MCLK ≀ ≀ 11111: 31MCLK

Note: Handshaking mode can only be set when (MCLK frequency/SYSCLK frequency) = 4.
If (MCLK frequency/SYSCLK frequency) = 1 or 2, set B3WM = 0 in MEMCTR3B.

After the reset is released, block 3 is set as follows:

Address output end timing	3MCLK
$\overline{\mathrm{RE}}$ negate timing	29MCLK
$\overline{\mathrm{WE}}$ negate timing	29MCLK
$\overline{\mathrm{RE}}$/$\overline{\mathrm{WE}}$ assert timing	3MCLK
Bus cycle start timing	0MCLK
Bus cycle end timing	31MCLK
AS assert timing	1MCLK
AS negate timing	3MCLK

The bus width is 16 bits, and synchronous fixed wait mode is set.

8.6.5 DRAM control register

DRAM control register

Register symbol: DRAMCTR
Address: x'32000040
Purpose: Stores various DRAM mode settings when DRAM is connected.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	–	–	–	–	RERS3	RERS2	RERS1	RERS0	SIZE1	SIZE0	–	–	BWC	REFE	PAGE	DRAME
Reset	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
Access	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W

Bit No.	Bit name	Description	Setting conditions
0	DRAME	DRAM control circuit enable	0: Disabled 1: Enabled
1	PAGE	Page mode enable	0: Disabled 1: Enabled
2	REFE	Refresh enable	0: Disabled 1: Enabled
3	BWC	Byte wide control	0: 2 $\overline{\text{WE}}$ control 1: 2 $\overline{\text{CAS}}$ control
7 to 6	SIZE1 to 0	DRAM size 00: Shift the address 9-bits to the low-order side and use as the row address 01: Shift the address 10-bits to the low-order side and use as the row address 10: Shift the address 11-bits to the low-order side and use as the row address 11: Shift the address 8-bits to the low-order side and use as the row address	
11 to 8	RERS3 to 0	Number of MCLK while $\overline{\text{RAS}}$ is asserted in the refresh cycle.	0000: prohibited 0001: 1MCLK ≀ ≀ 1111: 15MCLK

For details on the $\overline{\text{RAS}}$ hold time, the $\overline{\text{RAS}}$ precharge cycle, the $\overline{\text{CAS}}$ pulse width, the row address setup timing, the column address output timing, and the column address setup timing, refer to memory control registers 1A/B and 2A/B.

For the timing charts when using DRAM, refer to section 8.14, “External Memory Space Access (DRAM Spaces).”

8.6.6 Refresh count register

Register symbol: REFCNT
Address: x'32000042
Purpose: Sets the DRAM refresh interval when DRAM is connected.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	REFC 15	REFC 14	REFC 13	REFC 12	REFC 11	REFC 10	REFC 9	REFC 8	REFC 7	REFC 6	REFC 5	REFC 4	REFC 3	REFC 2	REFC 1	REFC 0
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit No.	Bit name	Description	Setting conditions
15 to 0	REFC 15 to 0	DRAM refresh interval	x'0000: 1 SYSCLK ≀ ≀ x'FFFF: 65536 SYSCLK

The refresh interval is the (REFCNT setting +1) multiplied by the SYSCLK cycle.
For the DRAM refresh timing, refer to section 8.14.4, “DRAM Refresh.”

8.6.7 Page Row Address Register

Page Row Address Register

Register symbol: PRAR
Address: x'32000044
Purpose: Sets the row address for DRAM software page mode.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	PRAR 15	PRAR 14	PRAR 13	PRAR 12	PRAR 11	PRAR 10	PRAR 9	PRAR 8	PRAR 7	PRAR 6	PRAR 5	PRAR 4	PRAR 3	PRAR 2	PRAR 1	PRAR 0
Reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

When DRAM software page mode is initiated (i.e., when the PE bit in memory control register 1B/2B is set to “1”), the contents of the page row address register are output as the row address.

Set the row address in the page row address register before initiating DRAM software page mode. The row address that is set at this point should have already been subjected to the shift operation in accordance with the DRAM size.

8.6.8 Clock Control Register

Clock Control Register

Register symbol: CKCTR
Address: x'32004000
Purpose: Sets the internal clock multiplier, etc.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name name	–	–	–	–	–	–	–	–	–	–	–	–	–	–	MCK 1	MCK 0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit No.	Bit name	Description	Setting conditions
1 to 0	MCK1 to 0	MCLK frequency setting for the input frequency	00: 1x 01: 2x 10: 4x 11: prohibited

For details, refer to Chapter 6, “Clock Generator.”

◎ Notes when switching the internal clock multiplier

Be aware of the following points when setting the clock control register CKCTR and changing the internal clock multiplier.

- If external memory is accessed immediately after setting the clock control register CKCTR, the multiplier for the internal clock MCLK may change in the middle of the access, resulting in a change in the external bus timing.

The internal clock multiplier changes during an external memory access in any of the following cases:

(1) When executing a program in internal instruction memory, the internal clock multiplier changes in the event of an access initiated by an external memory access instruction that came within seven instructions after an instruction that writes to the clock control register CKCTR.

(2) When executing a program in external memory, the internal clock multiplier changes in the event of either an instruction read or an access initiated by an external memory access instruction, when either came within three instructions after an instruction that writes to the clock control register CKCTR.

(3) When interrupt processing is generated immediately after writing CKCTR and the stack pointer is pointing to the external memory space, the internal clock multiplier changes for the first write access to the stack immediately after the interrupt is accepted.

Use either of the following methods in response to the above situations:

Method 1: In situations where it does not matter if the clock multiplier changes during the external memory access

Write CKCTR after setting the bus controller for the external memory access to a value that permits external memory access with any clock multiplier before or after overwriting CKCTR.

Method 2: In situations where the clock multiplier must not change during the external memory access (limited to memory extension mode)

Overwrite CKCTR through a program in internal instruction memory, and then place at least seven instructions that do not access external memory (for example, nop instructions) after the instruction that writes CKCTR. In addition, if there is a possibility that an interrupt request may be generated immediately after the CKCTR write operation, either set the stack pointer in internal data RAM beforehand, or else prohibit the acceptance of interrupts before writing CKCTR.

Note that method 2 cannot be used in processor mode; use method 1.

8.7 Space Partitioning

In extension memory mode (MMOD 1 to 0 = "LH"), the 1 GB memory space from x'80000000 to x'BFFFFFFF becomes external memory space; in processor mode (MMOD 1 to 0 = "HL"), the 2 GB memory space from x'40000000 to x'BFFFFFFF becomes external memory space. External memory space is partitioned into 4 blocks (block 0 to block 3). When any of these blocks are accessed, various signals (such as $\overline{\text{CSn}}$) corresponding to the block in question are output. Fig. 8-7-1 shows the address format for external memory accesses, and Fig. 8-7-2 shows the memory map.

In addition, if it is necessary to extend the external memory space, extension address A23 (dual-purpose pin that is shared with $\overline{\text{CS3}}$) can be used to extend the memory space. (There is no portion for address extension for block 3.)

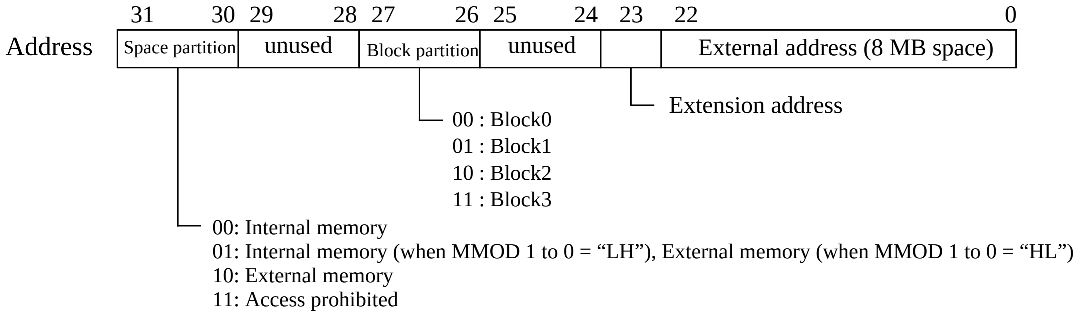


Fig. 8-7-1 Address Format When Accessing External Memory

In addition, the features of each block are described in Table 8-7-1. The various settings for each block are made through the memory block 0 to 3 control registers and, when DRAM is connected, through the DRAM control register.

Table 8-7-1 Features of Each Block

Block	DRAM connection	Bus width	Wait	Mode	Output signal used for block identification	Address data pins separate/multiplex
Block 0	Not permitted	8/16	Fixed	Synchronous /asynchronous	$\overline{\text{CS0}}$	Permitted
Block 1	Permitted	8/16	Fixed	Synchronous /asynchronous	$\overline{\text{CS1}}$ or $\overline{\text{RAS1}}$	Permitted
Block 2	Permitted	8/16	Fixed/handshaking	Synchronous /asynchronous	$\overline{\text{CS2}}$ or $\overline{\text{RAS2}}$	Permitted
Block 3	Not Permitted	8/16	Fixed/handshaking	Synchronous /asynchronous	$\overline{\text{CS3}}$	Permitted

For details, refer to the descriptions of memory control registers 0A/B, 1A/B, 2A/B, and 3A/B in section 8.6, “Description of Registers.”

Note 1. Handshaking can only be used in synchronous mode.
2. Connected DRAM can only be used when address and data are separate.

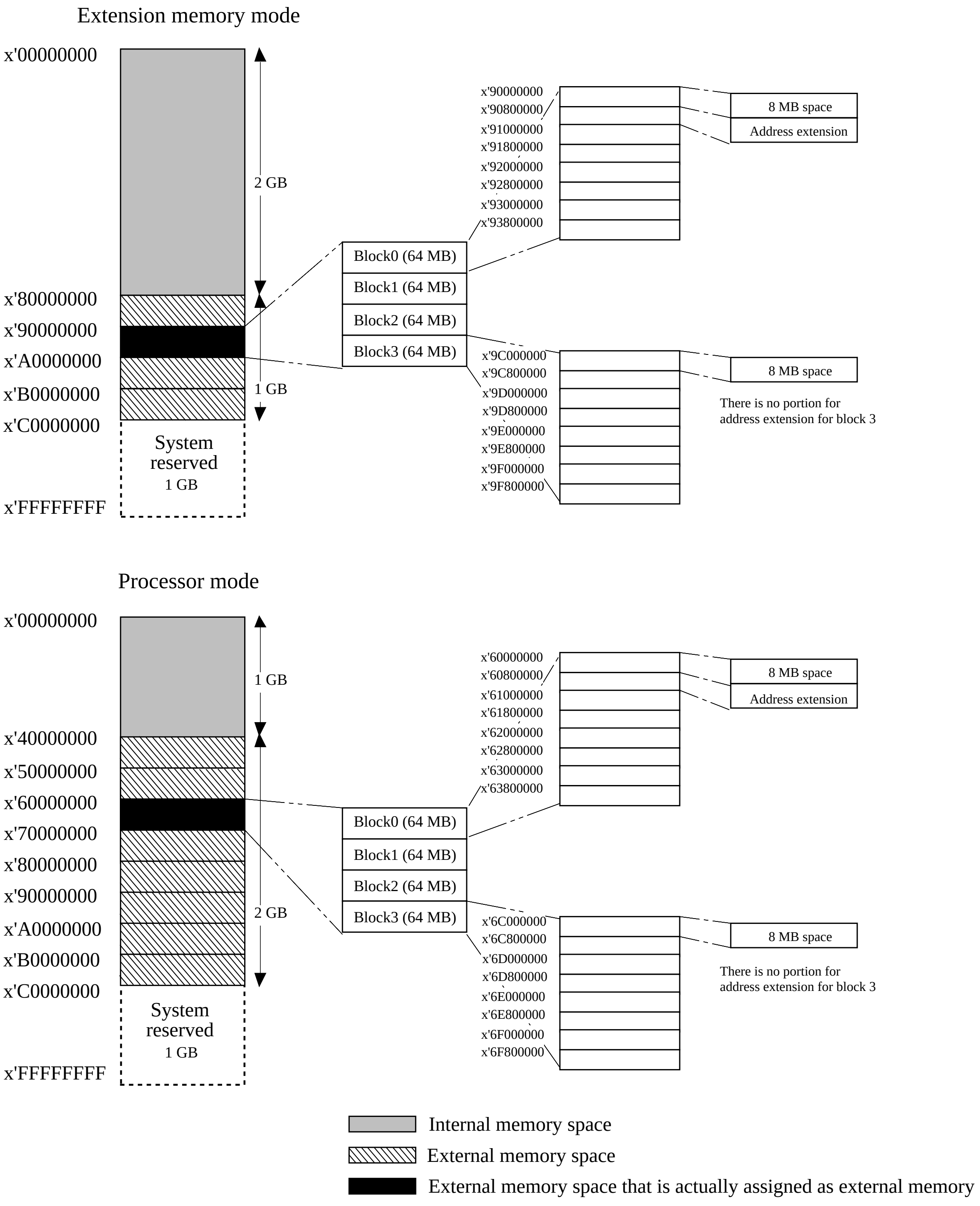


Fig. 8-7-2 Space Partitioning

8.8 Operation Clocks

MCLK, IOCLK, and SYSCLK are used as BC operation clocks. Table 8-8-1 shows the ratio of each clock versus the oscillation input clock (OSCI).

Table 8-8-1 Frequency Ratios of BC Operation Clocks

Oscillation mode		Clock control register setting	SYSCLK	MCLK	IOCLK
CKSEL	PLL		mutiplier	mutiplier	mutiplier
H	Using	MCK[1:0] = 10	1	4	1
		MCK[1:0] = 01		2	1/2
		MCK[1:0] = 00		1	1/4
L	Not using	Not using	1/2	1/2	1/8

For details, refer to Chapter 6, "Clock Generator."

8.9 Mode Settings

The values of external input pins MMOD1 to 0 and EXMOD1 to 0 set the external memory mode, block 0 bus width, and separate/common mode for the address pins and data pins. The various mode settings that can be made through the external pins are shown in Table 8-9-1.

Table 8-9-1 Mode Settings by the BC External Pins

Selection			Setting			
Mode name	Address/data	Block 0 bus width*	MMOD1	MMOD0	EXMOD1	EXMOD0
Memory extension mode	Separate	8 bits	L	H	L	L
		16 bits	L	H	L	H
	Common	8 bits	L	H	H	L
		16 bits	L	H	H	H
Processor mode	Separate	8 bits	H	L	L	L
		16 bits	H	L	L	H
	Common	8 bits	H	L	H	L
		16 bits	H	L	H	H

* Set the bus widths for blocks 1 to 3 through their respective memory control registers.

8.10 Bus Cycle

Depending on the value of the external input pin CKSEL and the internal registers, the MCLK frequency can be either 1/2, 1, 2, or 4 times the input frequency, and the IOCLK frequency can be either 1/8, 1/4, 1/2, or 1 times the input frequency. Note that SYSCLK is output with either 1/2 or 1 times the input frequency.

Table 8-10-1 Relationship between the Clock Frequency and the Number of Cycles (CPU Cycles) Required for Access

CKSEL			H			L
Clock control register MCK [1:0] = 11 is prohibited.			MCK [1:0] =10	MCK [1:0] = 01	MCK [1:0] = 00	(Not using)
Destination of access		MCLK/Input frequency	4	2	1	1/2
Internal instruction ROM/Internal flash memory	Instruction read		2	2	2	2
	Data read		3	3	3	3
Internal data RAM	Read/write		1	1	1	1
Control register in BC	Read		3	3	3	3
	Write		2	2	2	2
Internal I/O	Read	Synchronous	7 to 10	7 to 10	7 to 10	7 (*3)
	(*1) Write	Synchronous	6 to 9	6 to 9	6 to 9	(*3) 6
External memory	Read	(*2) Synchronous	Number of EX bus cycles + 3 to 6	Number of EX bus cycles + 3 to 4	Number of EX bus cycles + 3	Number of EX bus cycles + 3
		Asynchronous	Number of EX bus cycles + 3	Number of EX bus cycles + 3	Number of EX bus cycles + 3	Number of EX bus cycles + 3
	(*1) Write	(*2) Synchronous	Number of EX bus cycles + 2 to 5	Number of EX bus cycles + 2 to 3	Number of EX bus cycles + 2	Number of EX bus cycles + 2
		Asynchronous	Number of EX bus cycles +2	Number of EX bus cycles + 2	Number of EX bus cycles + 2	Number of EX bus cycles +2

(*1) If the store buffer is operational, the writing to internal I/O and external memory is entirely performed with 0 wait states.

(*2) In the synchronous mode, a synchronization wait of a maximum of 3 cycles or of 1 cycle is generated when the MCLK frequency is four times or two times the SYSCLK frequency, respectively.

(*3) Because the ratio of IOCLK to MCLK is always 1/4, a wait for synchronization is inserted.

8.11 Store Buffer

The bus controller has one store buffer (with a 32-bit data width) built in, and is used to avoid a time penalty when conducting a store operation in internal I/O or external memory. The CPU store operation is completed storing the address, data, and access size in the store buffer, and is executed with no wait states. Writes from the store buffer to internal I/O or external memory are conducted in parallel with subsequent CPU operations. However, if there is a request from the CPU for an access to the internal I/O or external memory before the write from the store buffer is completed, execution of that request is delayed.

8.12 Accessing the Internal I/O Space

Accesses to the internal I/O space (I/O register) are performed through the I/O bus, with the bus controller controlling the interface for read/write requests from the CPU. Accesses between the bus controller and the internal I/O space are executed in synchronization with IOCLK. Fig. 8-12-1 shows the timing chart when accessing the internal I/O space.

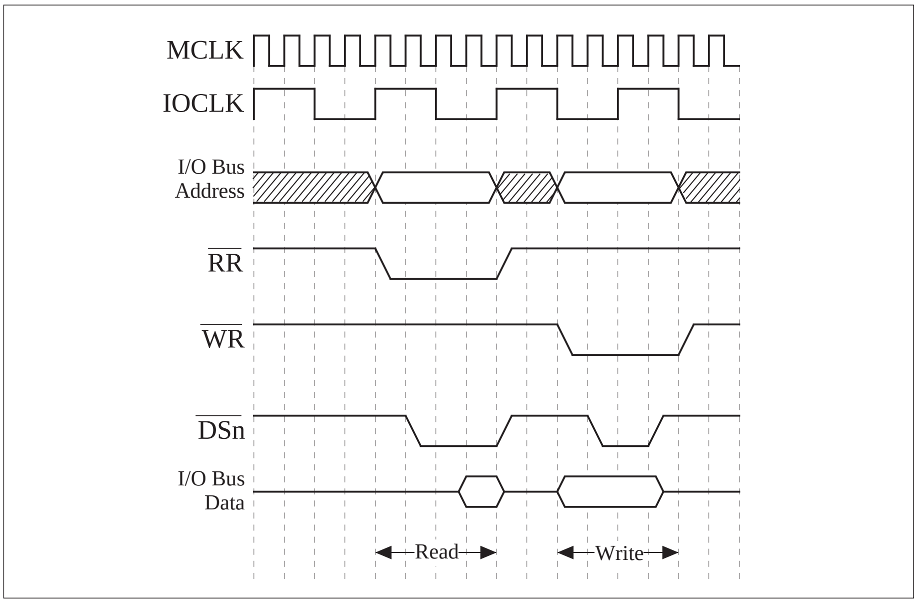


Fig. 8-12-1 Internal I/O Space Access

During a read, the address and the read request signal ($\overline{\text{RR}}$) are output in synchronization with the rising edge of IOCLK. After MCLK 1 cycle, the data strobe signals ($\overline{\text{DSn}}$) are asserted, and the I/O side begins to drive the data on the data bus. During a write, the address and the write request signal ($\overline{\text{WR}}$) are output in synchronization with the falling edge of IOCLK. After MCLK 1 cycle, the data strobe signals ($\overline{\text{DSn}}$) are asserted, and are then negated 1/4 of an IOCLK cycle before the end of the I/O access cycle. The write is performed at the rising edge of the $\overline{\text{DSn}}$ signals.

8.13 External Memory Space Access (Non-DRAM Spaces)

During an access to external memory, the BC controls the interface for the read/write request from the CPU. Table 8-13-1 lists the transactions that are supported for the external bus.

Table 8-13-1 External Bus Transaction

Address /data	Bus width	Mode		
		Synchronization		Asynchronization
Separation	8	Fixed wait	Handshaking	Fixed wait
	16	Fixed wait	Handshaking	Fixed wait
Multiplex	8	Fixed wait	Handshaking	Fixed wait
	16	Fixed wait	Handshaking	Fixed wait

8.13.1 16-bit Bus with Fixed Wait States, in Synchronous Mode and in Address/Data Separate Mode

Setting of the various parameters for external memory access is performed in memory control registers 0 to 3, corresponding to each block. In synchronous mode, the bus access is initiated in synchronization with SYSCLK. When fixed wait insertion is specified, the bus access ends to the timing set in the memory control register.
Fig. 8-13-1 is the timing chart in the case of a “16-bit bus with fixed wait states, in synchronous mode, in address/data separate mode, and with the frequency of MCLK equal to that of SYSCLK multiplied by four.”
Fig. 8-13-2 is the timing chart in the case of a “16-bit bus with fixed wait states, in synchronous mode, in address/data separate mode, and with the frequency of MCLK equal to that of SYSCLK multiplied by two.”
Fig. 8-13-3 is the timing chart in the case of a “16-bit bus with fixed wait states, in synchronous mode, in address/data separate mode, and with the frequency of MCLK equal to that of SYSCLK.”
BCS indicates the timing during one SYSCLK cycle at which the access should start, and is expressed in terms of the number of MCLK pulses since the rising edge of SYSCLK.
Note that when writing to byte 0, $\overline{\text{WE0}}$ is asserted and the data is output on D7 to 0, and when writing to byte 1, $\overline{\text{WE1}}$ is asserted and the data is output on D15 to 8.
In addition, in the case of a word access (32 bits), the external access is performed twice with A[1] = "0" and A[1] = "1".

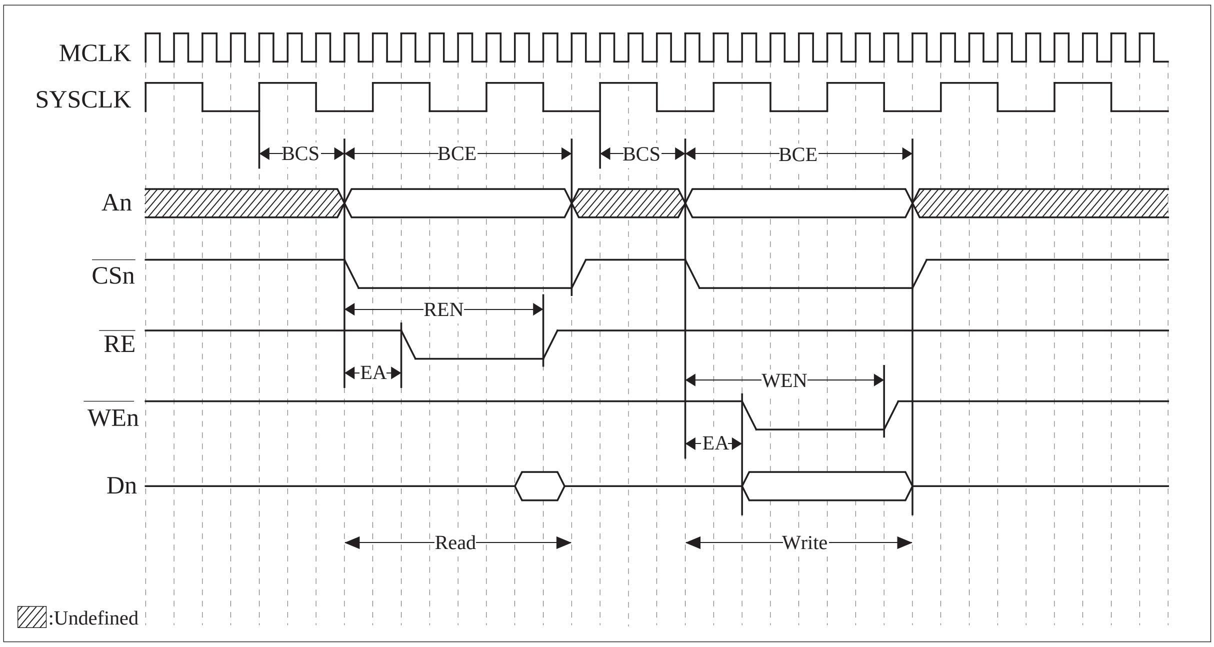


Fig. 8-13-1 Access Timing on a 16-bit Bus with Fixed Wait States, in Synchronous Mode and in Address/Data Separate Mode (MCLK = SYSCLK multiplied by 4)

For details on the various timing settings, refer to the description of the memory control register in section 8.6, “Description of Registers.”

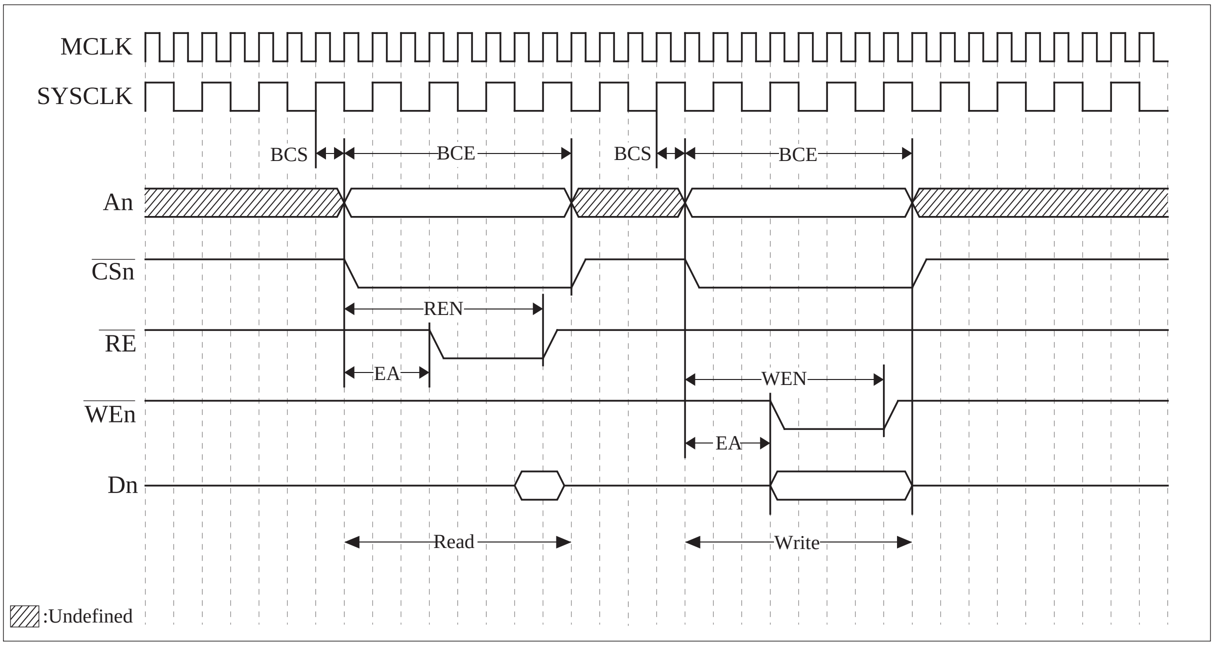


Fig. 8-13-2 Access Timing on a 16-bit Bus with Fixed Wait States, in Synchronous Mode and in Address/Data Separate Mode (MCLK = SYSCLK multiplied by 2)

For details on the various timing settings, refer to the description of the memory control register in section 8.6, “Description of Registers.”

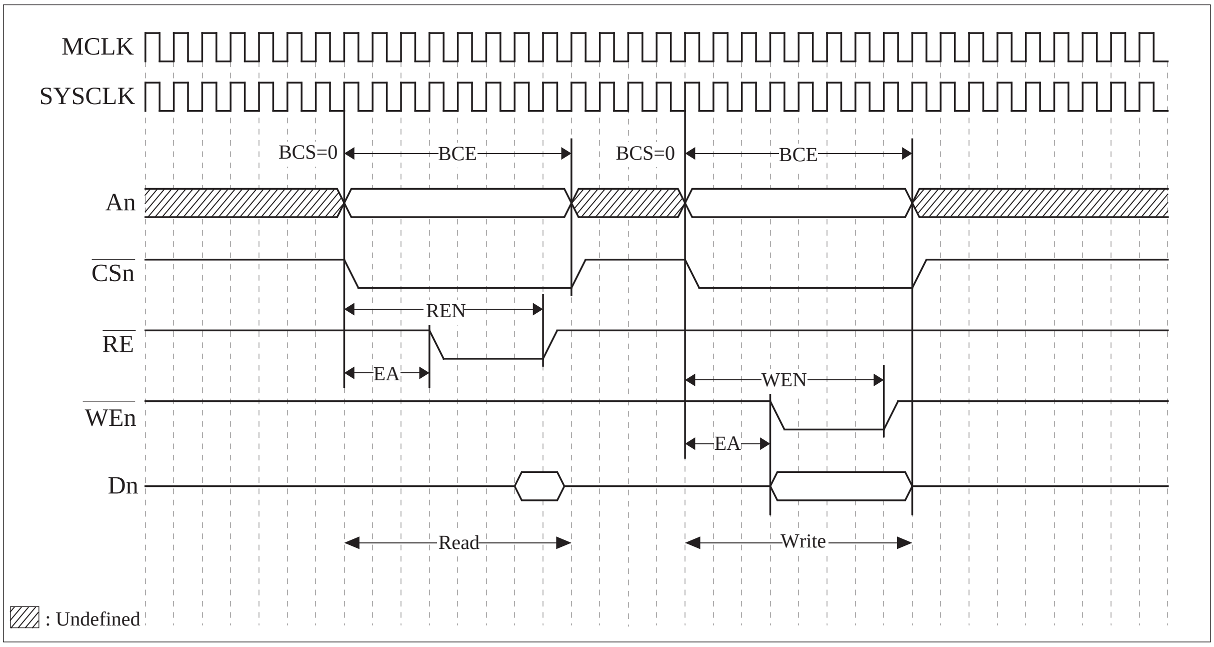


Fig. 8-13-3 Access Timing on a 16-bit Bus with Fixed Wait States, in Synchronous Mode and in Address/Data Separate Mode (MCLK = SYSCLK)

For details on the various timing settings, refer to the description of the memory control register in section 8.6, “Description of Registers.”

8.13.2 16-bit Bus with Handshaking, in Synchronous Mode and in Address/Data Separate Mode

When using handshaking, bus access starts once synchronization with SYSCLK is achieved, and after the data acknowledge signal ($\overline{\text{DK}}$) is asserted, 2 MCLK cycles are consumed by the BC internally and then the access is completed according to the specified parameters.

The various parameters for external memory access are set in memory control registers 2 and 3, corresponding to each block.

Handshaking can only be set in synchronous mode.

Fig. 8-13-4 is the timing chart in the case of a “16-bit bus with handshaking, in synchronous mode, in address/data separate mode, and with the frequency of MCLK equal to that of SYSCLK multiplied by four.”

Fig. 8-13-5 is the timing chart in the case of a “16-bit bus with handshaking, in synchronous mode, in address/data separate mode, and with the frequency of MCLK equal to that of SYSCLK multiplied by two.”

Fig. 8-13-6 is the timing chart in the case of a “16-bit bus with handshaking, in synchronous mode, in address/data separate mode, and with the frequency of MCLK equal to that of SYSCLK.”

The $\overline{\text{DK}}$ signal connected to the microcontroller should be input so as to be asserted from point EA+DW onward, and is negated before the next access.

Note that when writing to byte 0, $\overline{\text{WE0}}$ is asserted and the data is output on D7 to 0, and when writing to byte 1, $\overline{\text{WE1}}$ is asserted and the data is output on D15 to 8.

In addition, in the case of a word access (32 bits), the external access is performed twice with A[1] = "0" and A[1] = "1".

Note: Setting handshaking is prohibited if synchronous mode has not been set.

Note: If handshaking mode is set for memory block 3, the only settings that are permitted are those in which MCLK is equal to SYSCLK multiplied by 4. Any setting in which MCLK is only twice SYSCLK, or in which the two frequencies are equal, is prohibited.

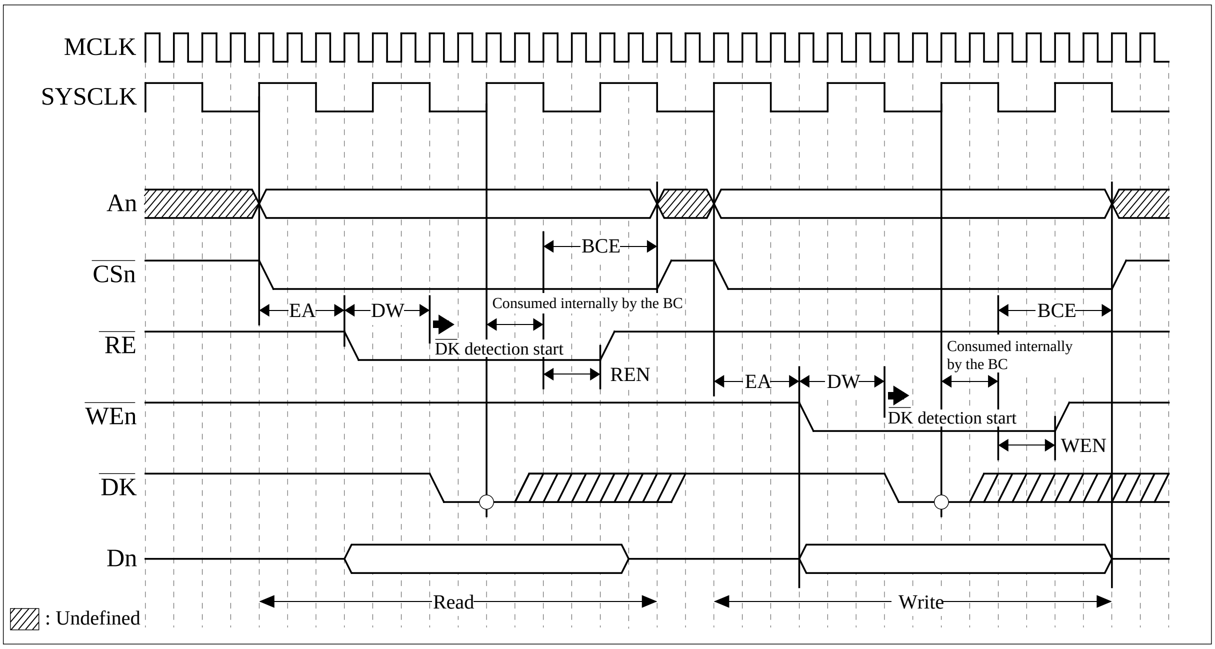


Fig. 8-13-4 Access Timing on a 16-bit Bus with Handshaking, in Synchronous Mode and in Address/Data Separate Mode (MCLK = SYSCLK multiplied by 4)

For details on the various timing settings, refer to the description of the memory control register in section 8.6, “Description of Registers.”

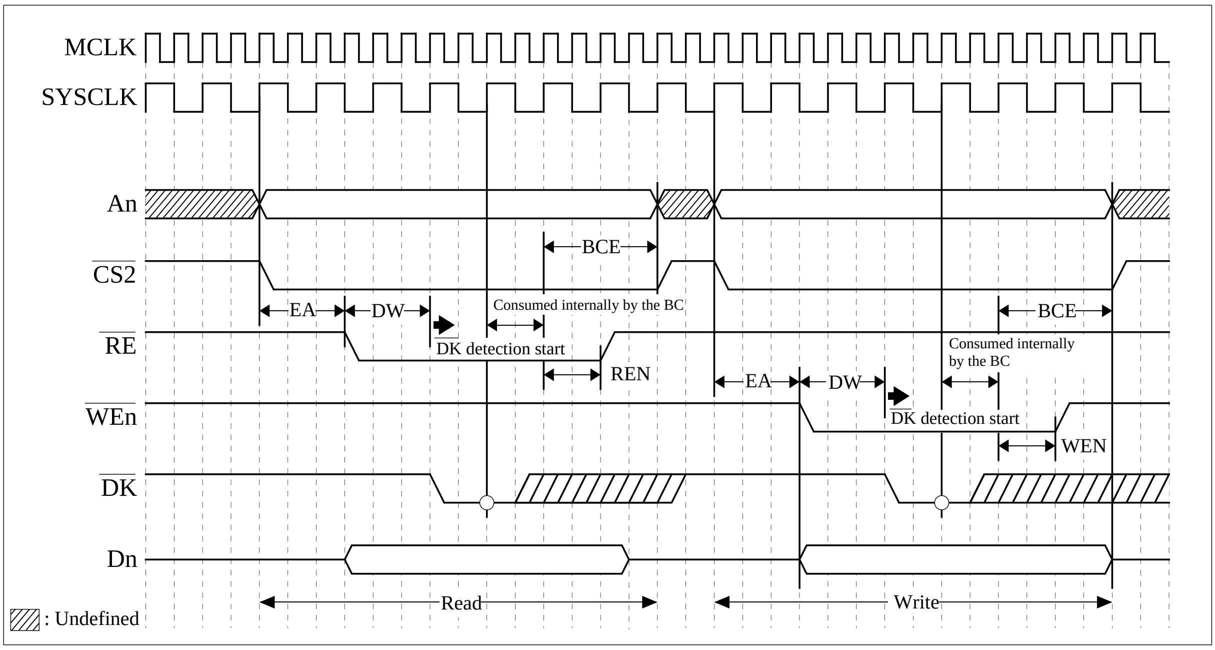


Fig. 8-13-5 Access Timing on a 16-bit Bus with Handshaking, in Synchronous Mode and in Address/Data Separate Mode (MCLK = SYSCLK multiplied by 2)

For details on the various timing settings, refer to the description of the memory control register in section 8.6, “Description of Registers.”

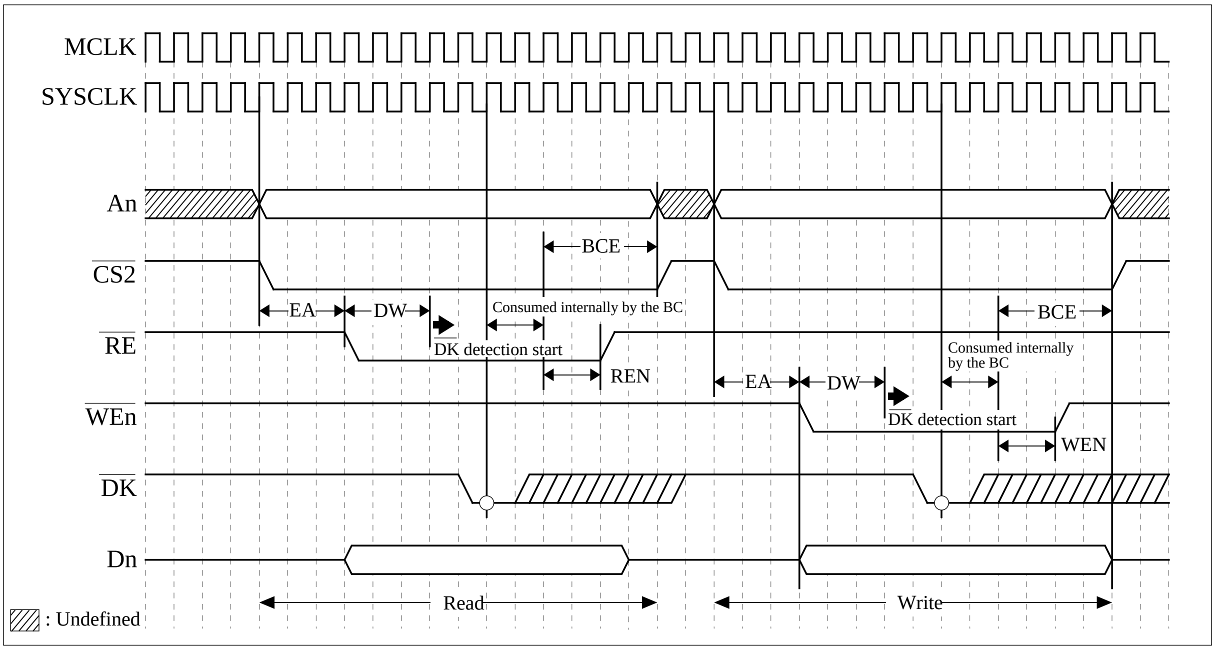


Fig. 8-13-6 Access Timing on a 16-bit Bus with Handshaking, in Synchronous Mode and in Address/Data Separate Mode (MCLK = SYSCLK)

For details on the various timing settings, refer to the description of the memory control register in section 8.6, “Description of Registers.”

8.13.3 16-bit Bus in Asynchronous Mode and in Address/Data Separate Mode

Asynchronous mode is used for accessing external memory at high speed; the address signals, $\overline{\text{CSn}}$ signals, etc., are output asynchronously with the SYSCLK but in synchronization with the internal MCLK. In asynchronous mode, accesses are all by fixed wait insertion.

Fig. 8-13-7 is the timing chart in the case of a “16-bit bus in asynchronous mode, in address/data separate mode, and with the frequency of MCLK equal to that of SYSCLK multiplied by four.”

Fig. 8-13-8 is the timing chart in the case of a “16-bit bus in asynchronous mode, in address/data separate mode, and with the frequency of MCLK equal to that of SYSCLK multiplied by two.”

Fig. 8-13-9 is the timing chart in the case of a “16-bit bus in asynchronous mode, in address/data separate mode, and with the frequency of MCLK equal to that of SYSCLK.”

During a read, the $\overline{\text{RE}}$ signal is asserted at EA x MCLK after the start of the bus cycle. During a write, the $\overline{\text{WE}}$ signal is asserted at EA x MCLK after the start of the bus cycle.

Note that when writing to byte 0, $\overline{\text{WE0}}$ is asserted and the data is output on D7 to 0, and when writing to byte 1, $\overline{\text{WE1}}$ is asserted and the data is output on D15 to 8.

In addition, in the case of a word access (32 bits), the external access is performed twice with A[1] = "0" and A[1] = "1".

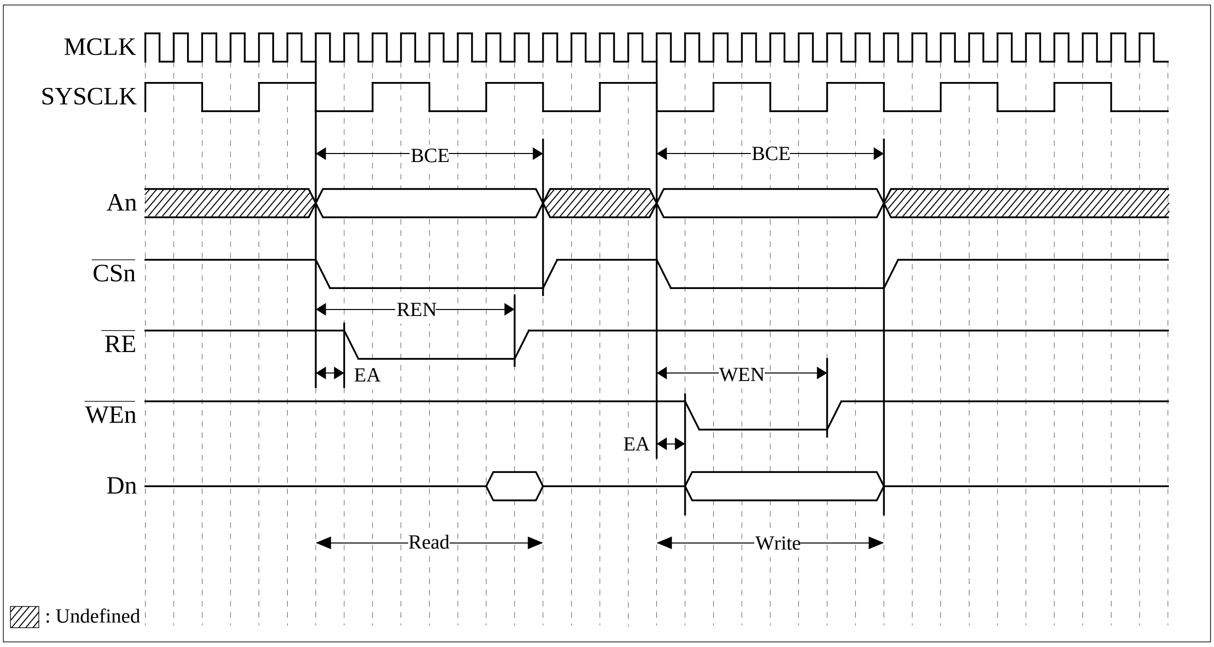


Fig. 8-13-7 Access Timing on a 16-bit Bus in Asynchronous Mode and in Address/Data Separate Mode (MCLK = SYSCLK multiplied by 4)

For details on the various timing settings, refer to the description of the memory control register in section 8.6, “Description of Registers.”

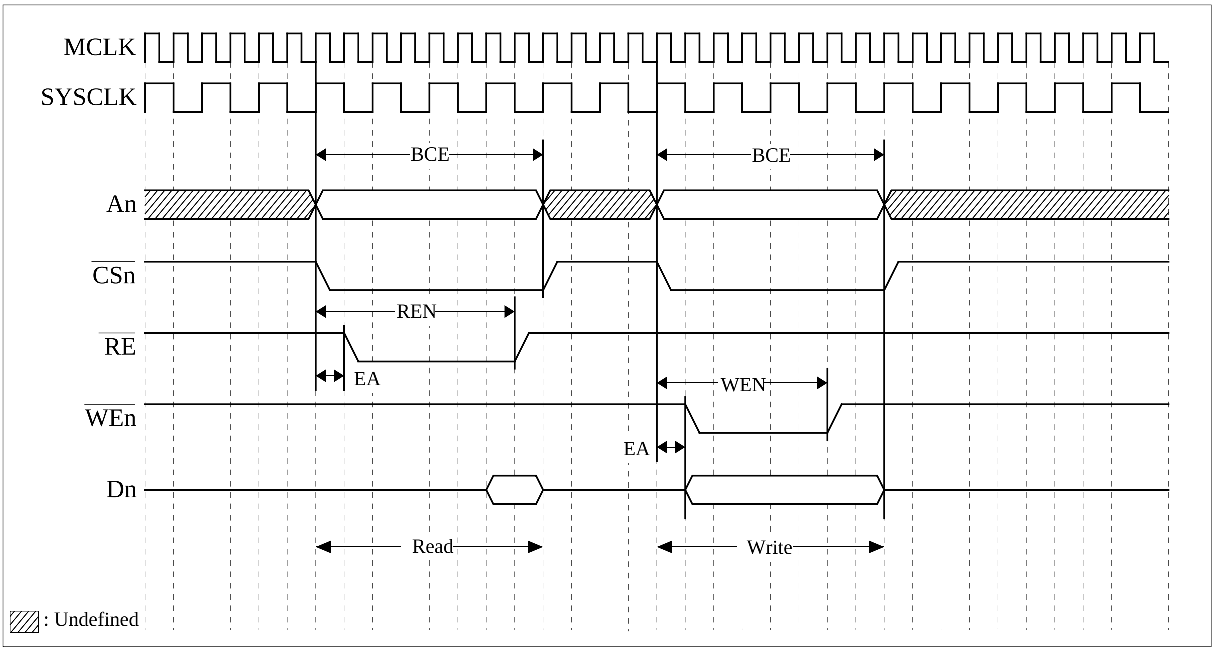


Fig. 8-13-8 Access Timing on a 16-bit Bus in Asynchronous Mode and in Address/Data Separate Mode (MCLK = SYSCLK multiplied by 2)

For details on the various timing settings, refer to the description of the memory control register in section 8.6, “Description of Registers.”

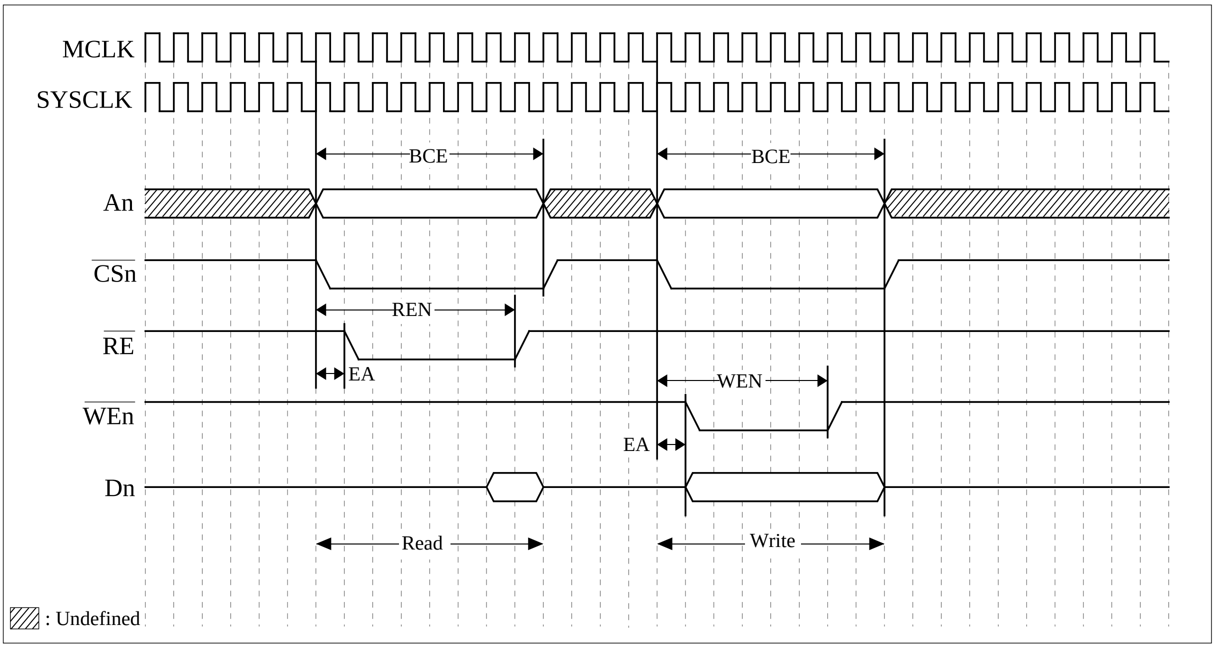


Fig. 8-13-9 Access Timing on a 16-bit Bus in Asynchronous Mode and in Address/Data Separate Mode (MCLK = SYSCLK)

For details on the various timing settings, refer to the description of the memory control register in section 8.6, “Description of Registers.”

8.13.4 8-bit Bus with Fixed Wait States, in Synchronous Mode and in Address/Data Separate Mode

8-bit bus mode is set for block 0 by setting the mode through the MMOD1 and 0 pins and the EXMOD1 and 0 pins, and for blocks 1 to 3 by setting the BnBW bit to "0" in the corresponding memory control register. In 8-bit bus mode, half-word access (16 bits) is performed by means of two external accesses, with A[0] = "0" for the low-order byte and A[0] = "1" for the high-order byte. Word access (32 bits) is performed by means of four accesses, with A[1:0] = "00", A[1:0] = "01", A[1:0] = "10", and A[1:0] = "11", starting from the low-order side. Note that the low-order 8 bits (D7 to 0) are used for the data bus.

In synchronous mode, the bus access starts in synchronization with SYSCLK, and when fixed wait states are inserted, the access ends according to the timing that was set in the memory control register.

The various parameters for external memory access are set in memory control registers 0 to 3, corresponding to each block.

Fig. 8-13-10 is the timing chart in the case of a half-word access using an “8-bit bus with fixed wait states, in synchronous mode, in address/data separate mode, and with the frequency of MCLK equal to that of SYSCLK multiplied by four.”

Fig. 8-13-11 is the timing chart in the case of a half-word access using an “8-bit bus with fixed wait states, in synchronous mode, in address/data separate mode, and with the frequency of MCLK equal to that of SYSCLK multiplied by two.”

Fig. 8-13-12 is the timing chart in the case of a half-word access using an “8-bit bus with fixed wait states, in synchronous mode, in address/data separate mode, and with the frequency of MCLK equal to that of SYSCLK.”

Note that when writing, $\overline{\text{WE0}}$ is asserted and the data is output on D7 to 0.

Note: For details on the mode settings, refer to Table 8-9-1, “Mode Settings by the BC External Pins.”

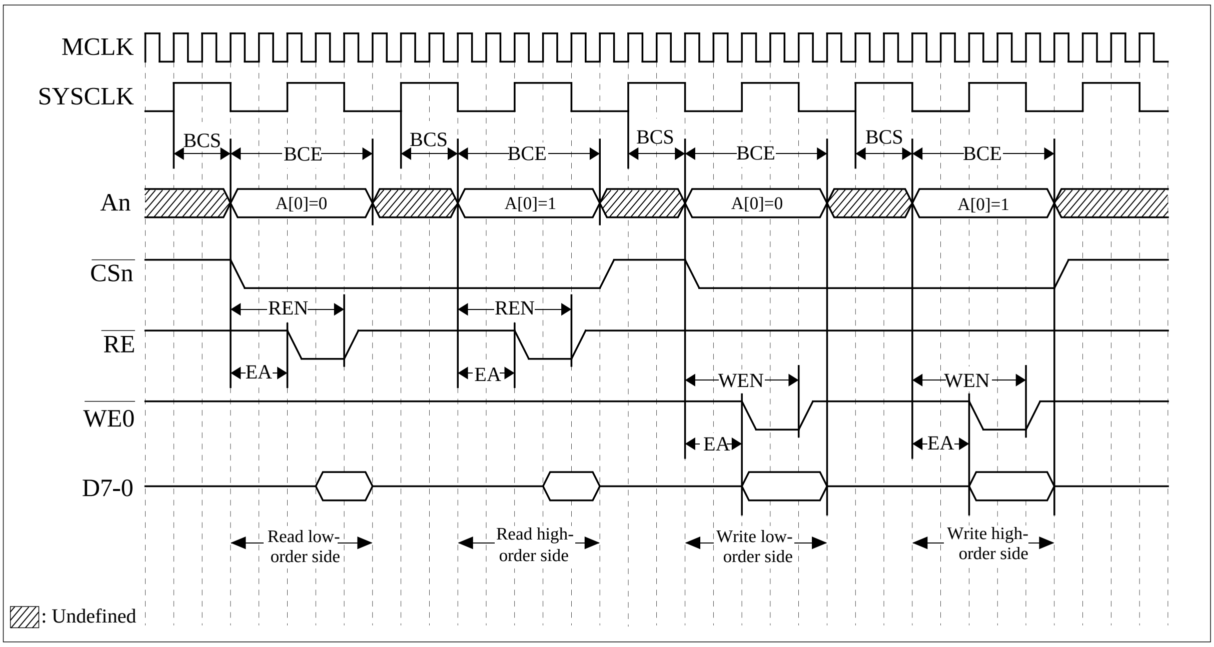


Fig. 8-13-10 Access Timing on a 8-bit Bus with Fixed Wait States, in Synchronous Mode and in Address/Data Separate Mode (MCLK = SYSCLK multiplied by 4)

For details on the various timing settings, refer to the description of the memory control register in section 8.6, “Description of Registers.”

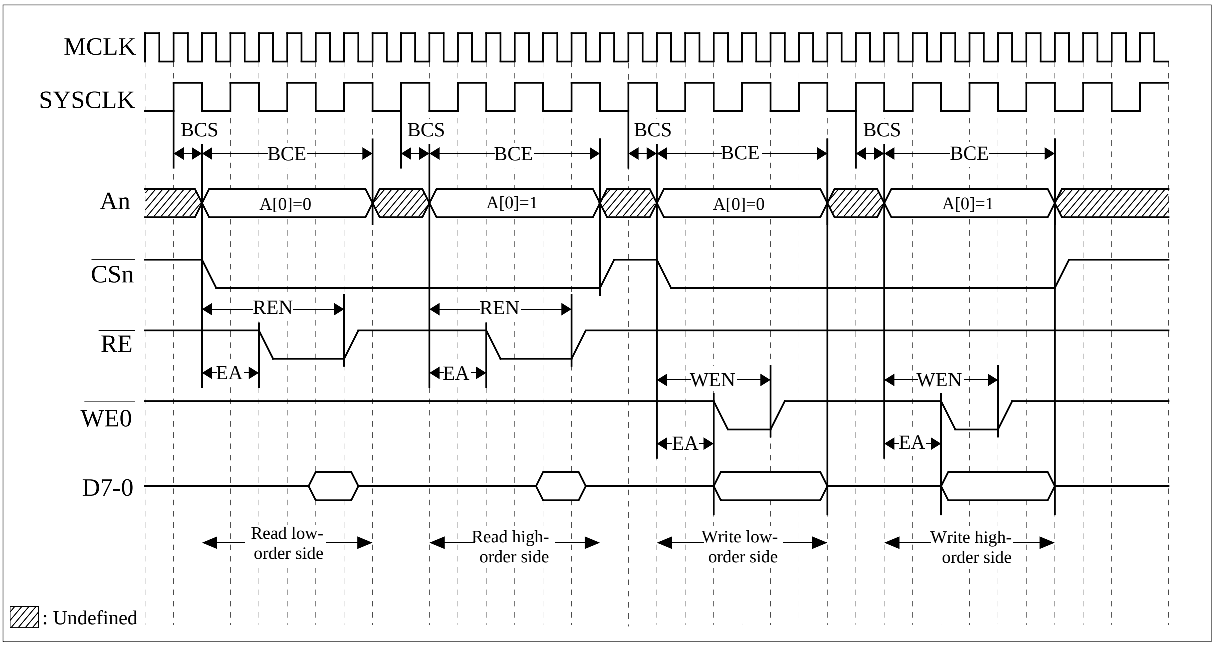


Fig. 8-13-11 Access Timing on a 8-bit Bus with Fixed Wait States, in Synchronous Mode and in Address/Data Separate Mode (MCLK = SYSCLK multiplied by 2)

For details on the various timing settings, refer to the description of the memory control register in section 8.6, “Description of Registers.”

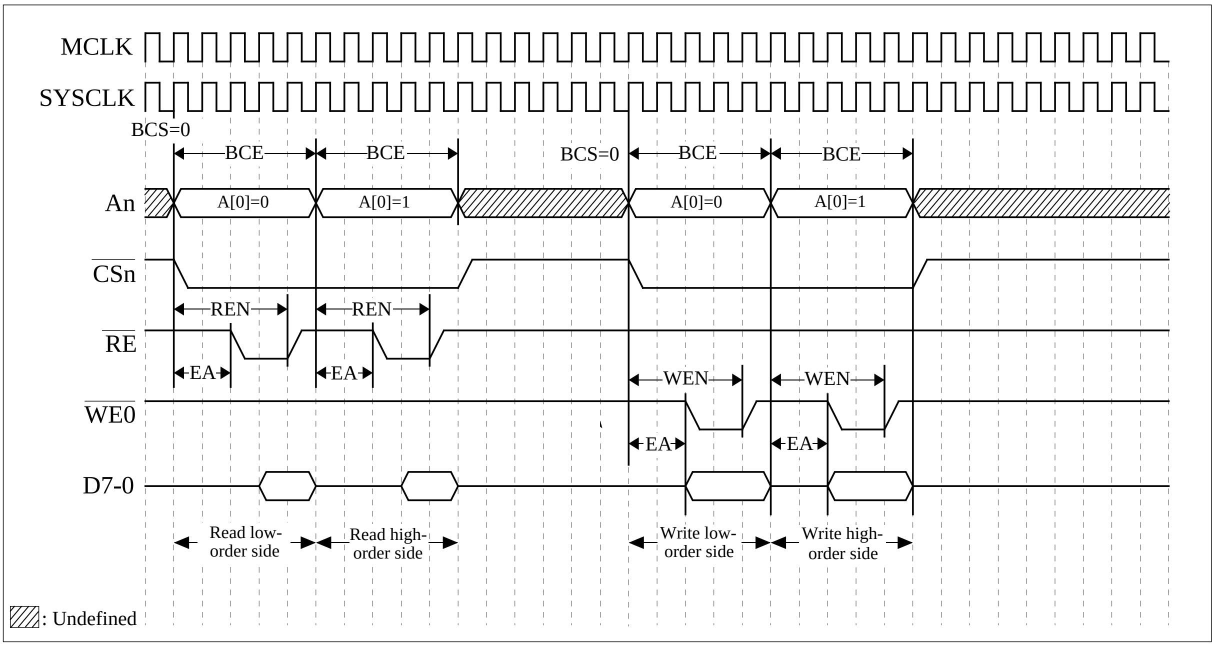


Fig. 8-13-12 Access Timing on a 8-bit Bus with Fixed Wait States, in Synchronous Mode and in Address/Data Separate Mode (MCLK = SYSCLK)

For details on the various timing settings, refer to the description of the memory control register in section 8.6, “Description of Registers.”

8.13.5 8-bit Bus with Handshaking, in Synchronous Mode and in Address/Data Separate Mode

8-bit bus mode is set for blocks 2 and 3 by setting the BnBW bit to “0” in the corresponding memory control register. In 8-bit bus mode, half-word access (16 bits) is performed by means of two external accesses, with A[0] = "0" for the low-order byte and A[0] = "1" for the high-order byte. Word access (32 bits) is performed by means of four accesses, with A[1:0] = "00", A[1:0] = "01", A[1:0] = "10", and A[1:0] = "11", starting from the low-order side. Note that the low-order 8 bits (D7 to 0) are used for the data bus.

When using handshaking, bus access starts once synchronization with SYSCLK is achieved, and after the data acknowledge signal ($\overline{\text{DK}}$) is asserted, 2 MCLK cycles are consumed by the BC internally and then the access is completed according to the specified parameters.

The various parameters for external memory access are set in memory control registers 2 and 3, corresponding to each block.

Handshaking can only be set in synchronous mode.

Fig. 8-13-13 is the timing chart in the case of a half-word access using an “8-bit bus with handshaking, in synchronous mode, in address/data separate mode, and with the frequency of MCLK equal to that of SYSCLK multiplied by four.”

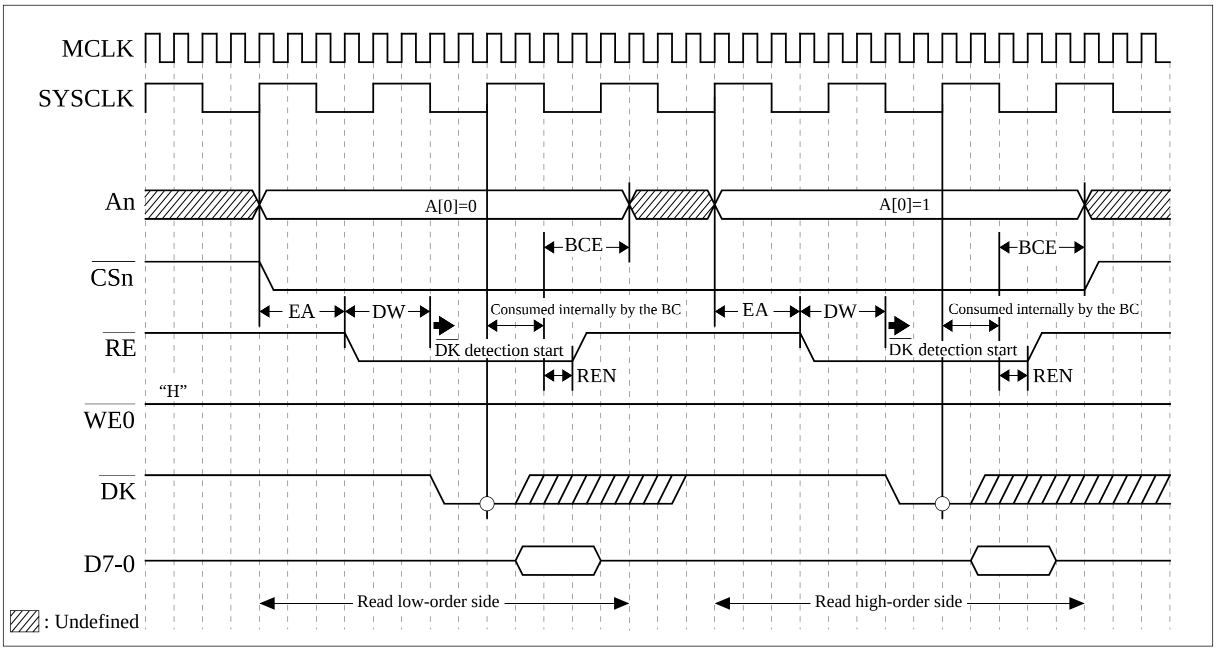
Fig. 8-13-14 is the timing chart in the case of a half-word access using an “8-bit bus with handshaking, in synchronous mode, in address/data separate mode, and with the frequency of MCLK equal to that of SYSCLK multiplied by two.”

Fig. 8-13-15 is the timing chart in the case of a half-word access using an “8-bit bus with handshaking, in synchronous mode, in address/data separate mode, and with the frequency of MCLK equal to that of SYSCLK.”

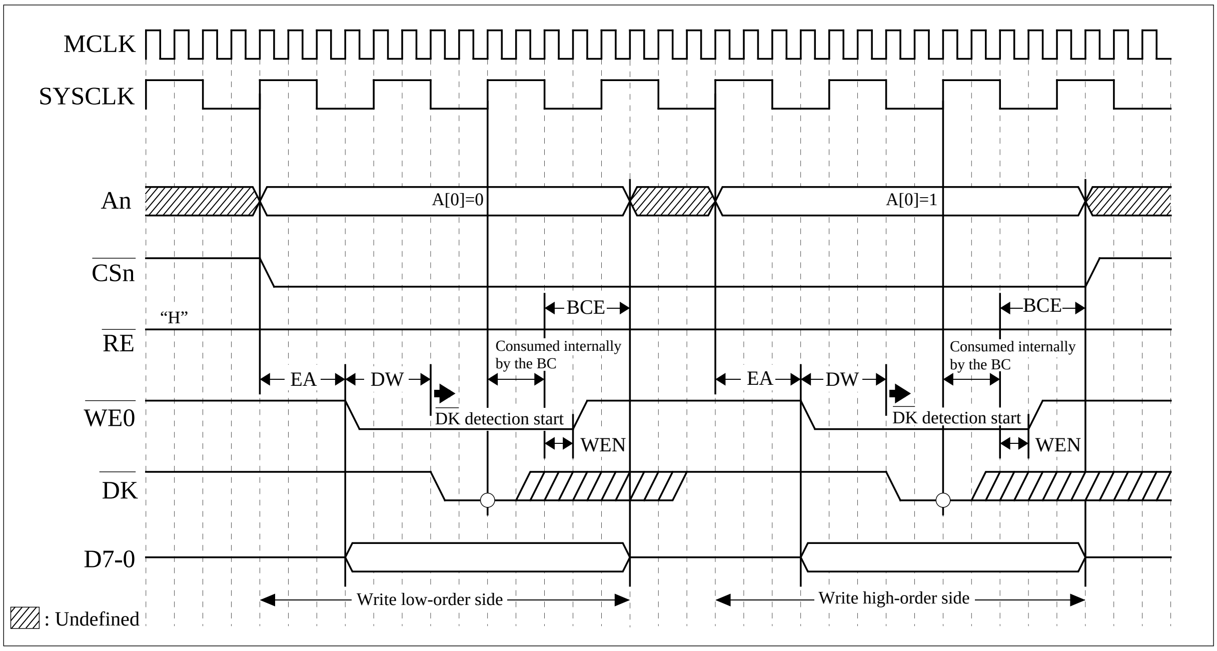
The $\overline{\text{DK}}$ signal connected to the microcontroller should be input so as to be asserted from point EA+DW onward, and is negated before the next access.

Note that when writing, $\overline{\text{WE0}}$ is asserted and the data is output on D7 to 0.

Note: If handshaking mode is set for memory block 3, the only settings that are permitted are those in which MCLK is equal to SYSCLK multiplied by 4. Any setting in which MCLK is only twice SYSCLK, or in which the two frequencies are equal, is prohibited.



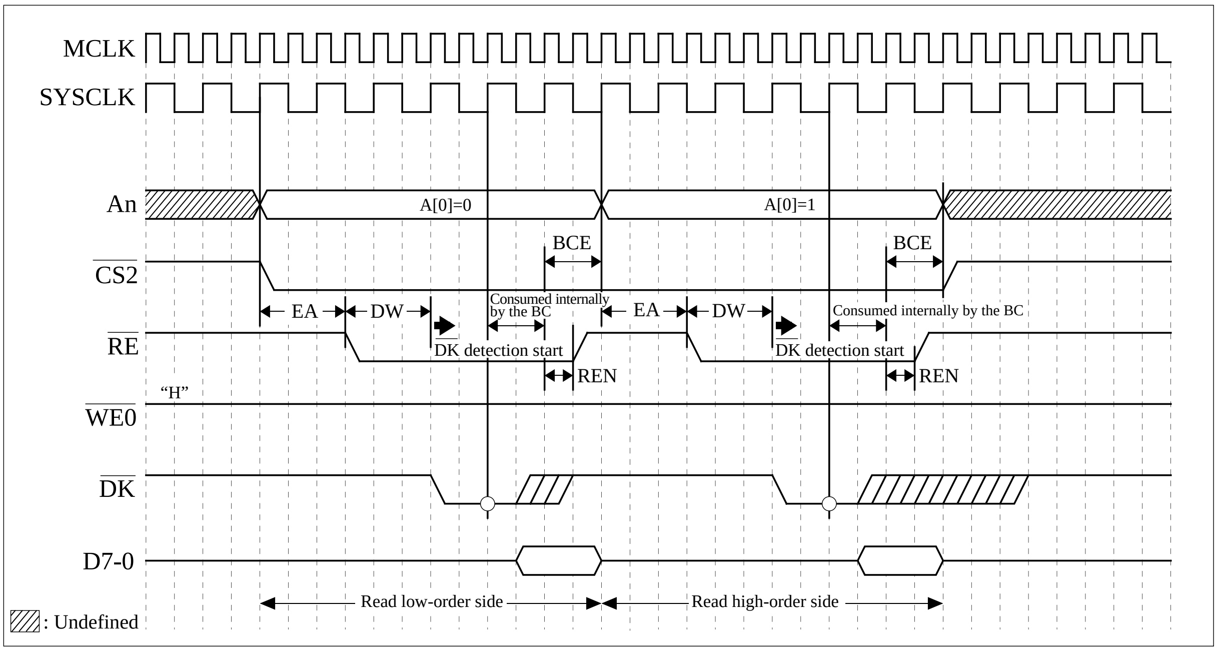
(a) Read Timing



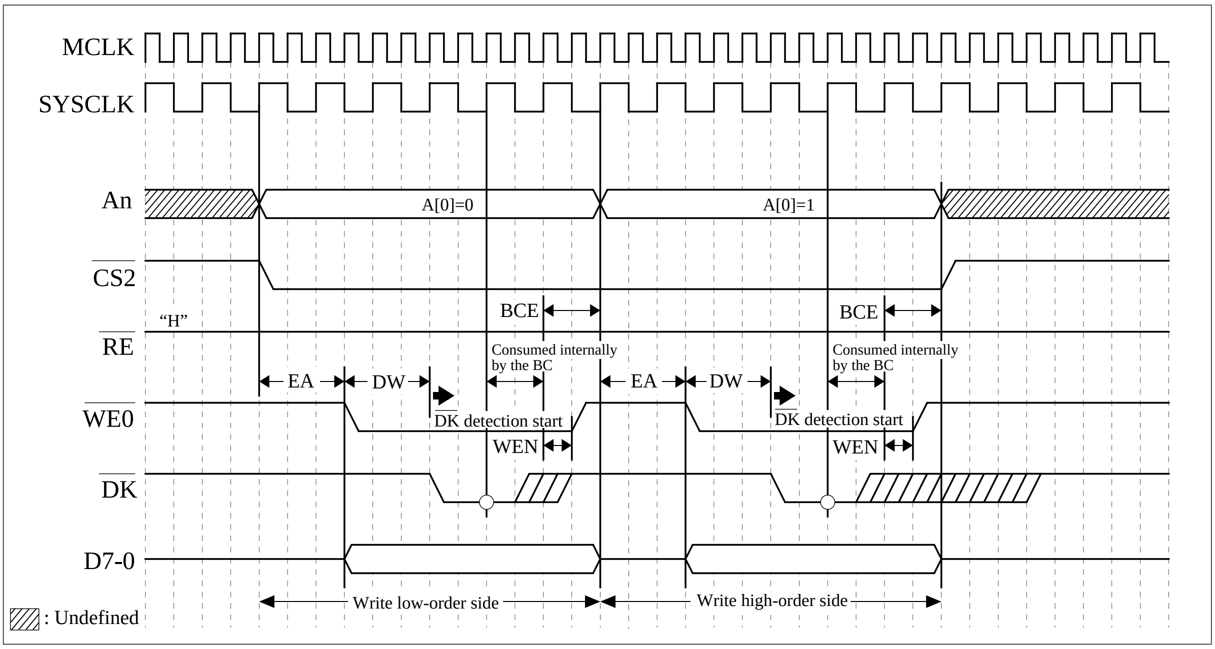
(b) Write Timing

Fig. 8-13-13 Access Timing on a 8-bit Bus with Handshaking, in Synchronous Mode and in Address/Data Separate Mode (MCLK = SYSCLK multiplied by 4)

For details on the various timing settings, refer to the description of the memory control register in section 8.6, "Description of Registers."



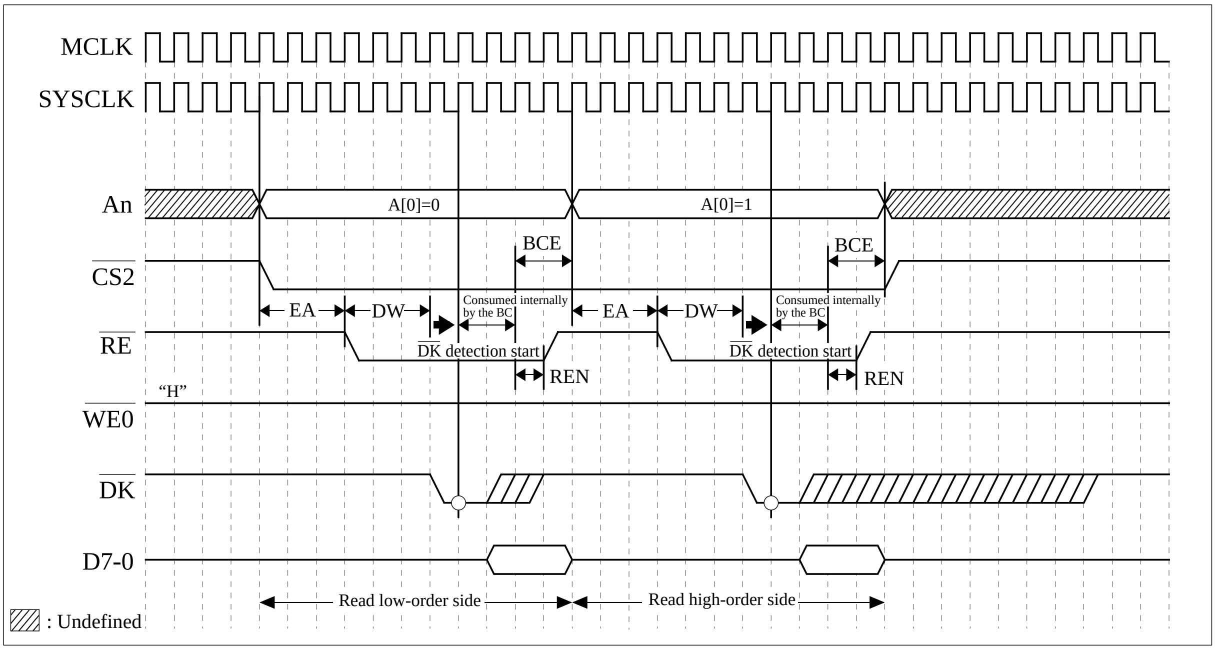
(a) Read Timing



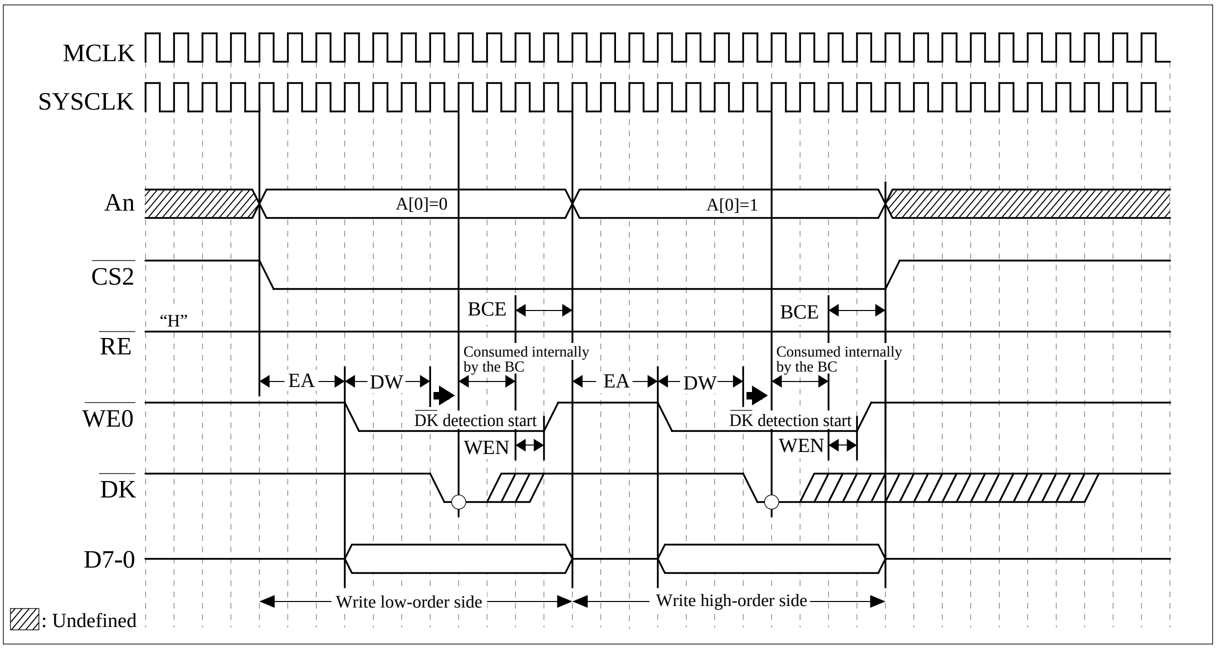
(b) Write Timing

Fig. 8-13-14 Access Timing on a 8-bit Bus with Handshaking, in Synchronous Mode and in Address/Data Separate Mode (MCLK = SYSCLK multiplied by 2)

For details on the various timing settings, refer to the description of the memory control register in section 8.6, "Description of Registers."



(a) Read Timing



(b) Write Timing

Fig. 8-13-15 Access Timing on a 8-bit Bus with Handshaking, in Synchronous Mode and in Address/Data Separate Mode (MCLK = SYSCLK)

For details on the various timing settings, refer to the description of the memory control register in section 8.6, "Description of Registers."

8.13.6 8-bit Bus in Asynchronous Mode and in Address/Data Separate Mode

8-bit bus mode is set for block 0 by setting the mode through the MMOD1 and 0 pins and the EXMOD1 and 0 pins, and for blocks 1 to 3 by setting the BnBW bit to “0” in the corresponding memory control register. In 8-bit bus mode, half-word access (16 bits) is performed by means of two external accesses, with A[0] = "0" for the low-order byte and A[0] = "1" for the high-order byte. Word access (32 bits) is performed by means of four accesses, with A[1:0] = "00", A[1:0] = "01", A[1:0] = "10", and A[1:0] = "11", starting from the low-order side. Note that the low-order 8 bits (D7 to 0) are used for the data bus.
Asynchronous mode is used for accessing external memory at high speed; the address signals, $\overline{\text{CS}}$ signals, etc., are output asynchronously with SYSCLK but in synchronization with the internal MCLK. In asynchronous mode, accesses are all by fixed wait insertion.
Fig. 8-13-16 is the timing chart in the case of a half-word access using an “8-bit bus in asynchronous mode, in address/data separate mode.”
Note that when writing, $\overline{\text{WE0}}$ is asserted and the data is output on D7 to 0.

Note: For details on the mode settings, refer to Table 8-9-1, “Mode Settings by the BC External Pins.”

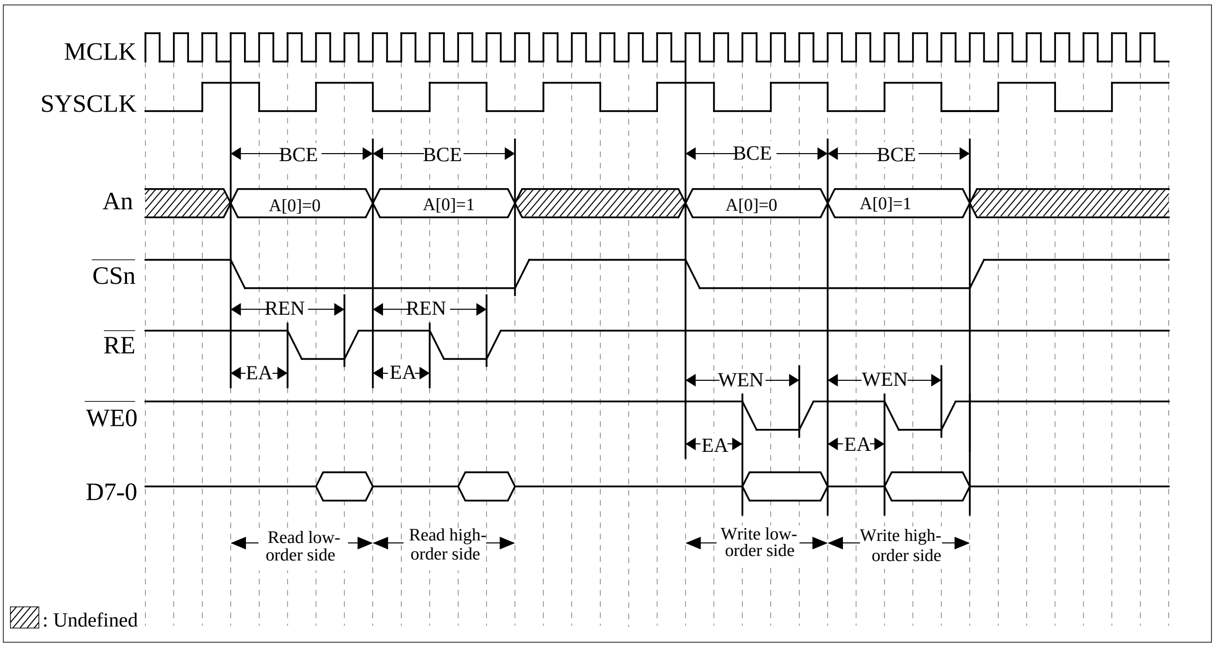


Fig. 8-13-16 Access Timing on a 8-bit Bus, in Asynchronous Mode and in Address/Data Separate Mode (MCLK = SYSCLK multiplied by 4)

For details on the various timing settings, refer to the description of the memory control register in section 8.6, “Description of Registers.”

8.13.7 16-bit Bus with Fixed Wait States, in Synchronous Mode and in Address/Data Multiplex Mode

By setting the mode through the MMOD1 and 0 pins and the EXMOD1 and 0 pins, blocks 0 to 3 use common pins for the memory address and memory data signals (pins ADM15 to 0). In synchronous mode, the bus access starts in synchronization with SYSCLK, and when fixed wait states are inserted, the access ends according to the timing that was set in the memory control register. The various parameters for external memory access are set in memory control registers 0 to 3, corresponding to each block.

BCS indicates the timing during one SYSCLK cycle at which the access should start, and is expressed in terms of the number of MCLK pulses since the rising edge of SYSCLK.

Fig. 8-13-17 is the timing chart in the case of a "16-bit bus with fixed wait states, in synchronous mode, in address/data multiplex mode, and with the frequency of MCLK equal to that of SYSCLK multiplied by four."

Fig. 8-13-18 is the timing chart in the case of a "16-bit bus with fixed wait states, in synchronous mode, in address/data multiplex mode, and with the frequency of MCLK equal to that of SYSCLK multiplied by two."

Fig. 8-13-19 is the timing chart in the case of a "16-bit bus with fixed wait states, in synchronous mode, in address/data multiplex mode, and with the frequency of MCLK equal to that of SYSCLK."

As shown in each timing chart, the ADM15 to 0 pins go to "Hi-Z" or the undefined output state while $\overline{\text{CSn}}$ is negated in address/data multiplex mode. When the bus authority is released, the ADM15 to 0 pins are either pulled up or go to "Hi-Z", depending on the setting of the I/O port output mode register.

Note that when writing to byte 0, $\overline{\text{WE0}}$ is asserted and the data is output on ADM7 to 0, and when writing to byte 1, $\overline{\text{WE1}}$ is asserted and the data is output on ADM15 to 8.

Note: For details on the mode settings, refer to Table 8-9-1, "Mode Settings by the BC External Pins."

Note: "0" (low level) is output on pins A23 to 16 (A23 also serves as $\overline{\text{CS3}}$) while the ADM15 to 0 pins function as data pins. Therefore, refer to 3. in section 8.16, "Cautions," regarding the use of these pins.

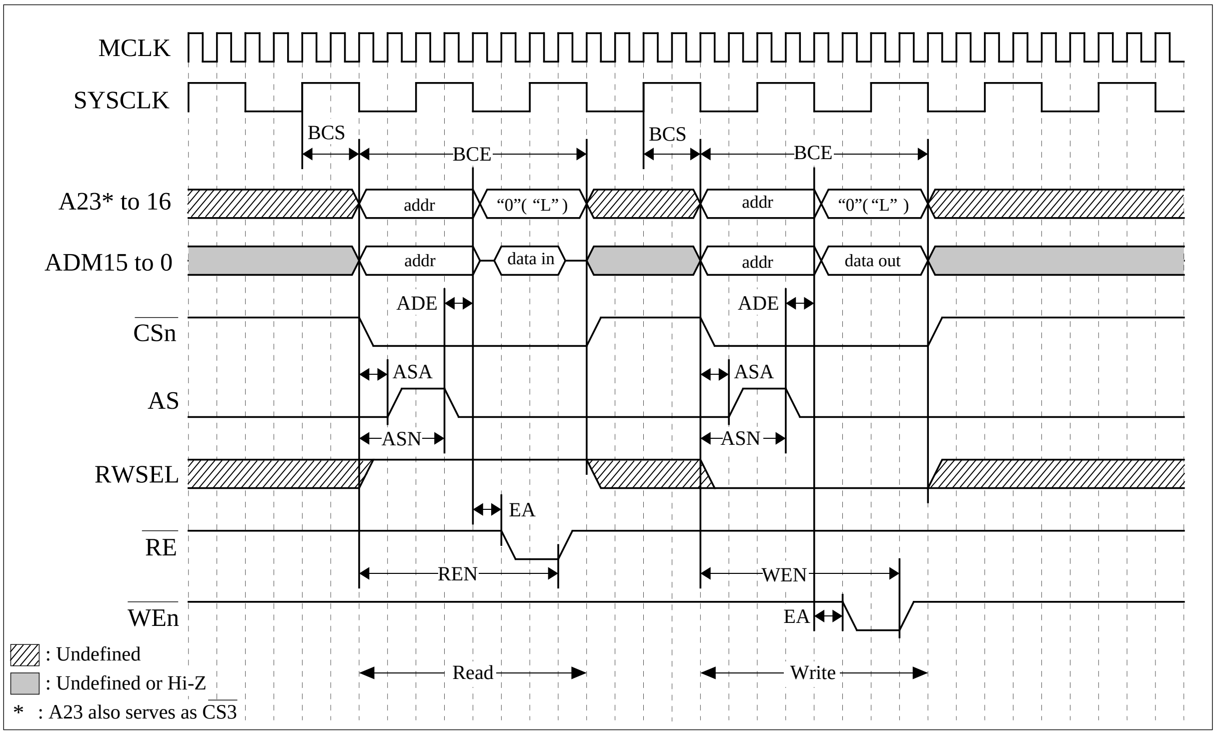


Fig. 8-13-17 Access Timing on a 16-bit Bus with Fixed Wait States, in Synchronous Mode and in Address/Data Multiplex Mode (MCLK = SYSCLK multiplied by 4)

For details on the various timing settings, refer to the description of the memory control register in section 8.6, "Description of Registers."

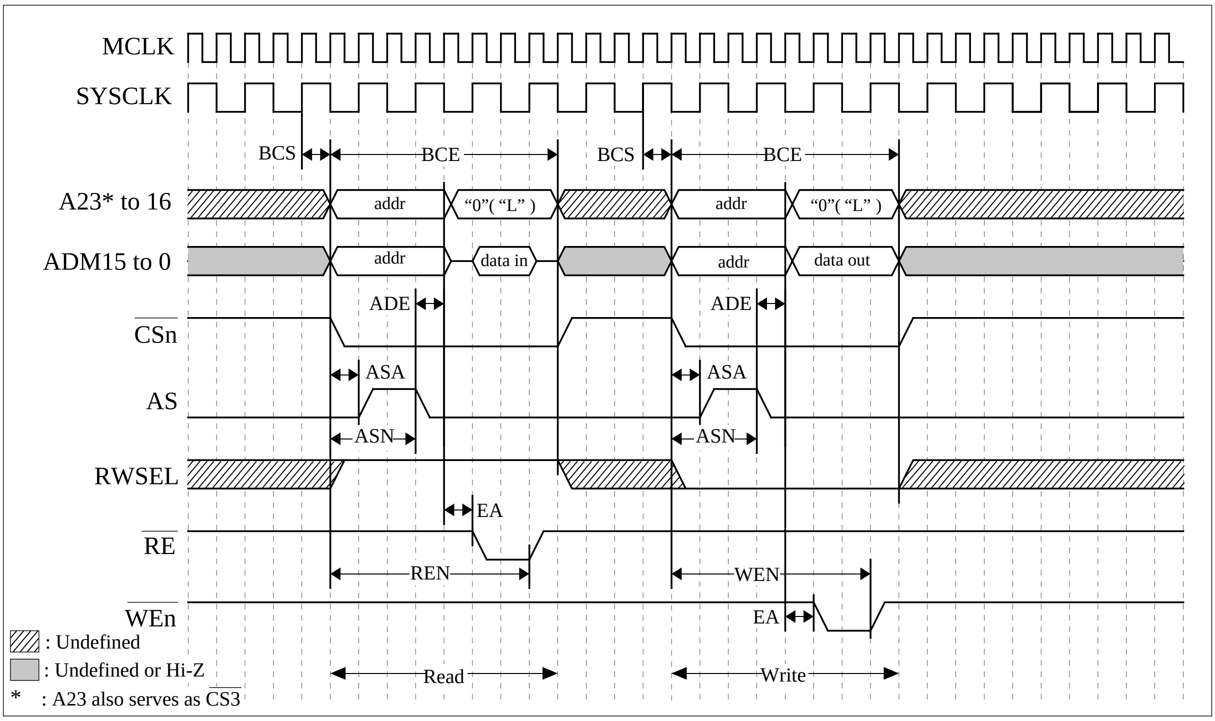


Fig. 8-13-18 Access Timing on a 16-bit Bus with Fixed Wait States, in Synchronous Mode and in Address/Data Multiplex Mode (MCLK = SYSCLK multiplied by 2)

For details on the various timing settings, refer to the description of the memory control register in section 8.6, "Description of Registers."

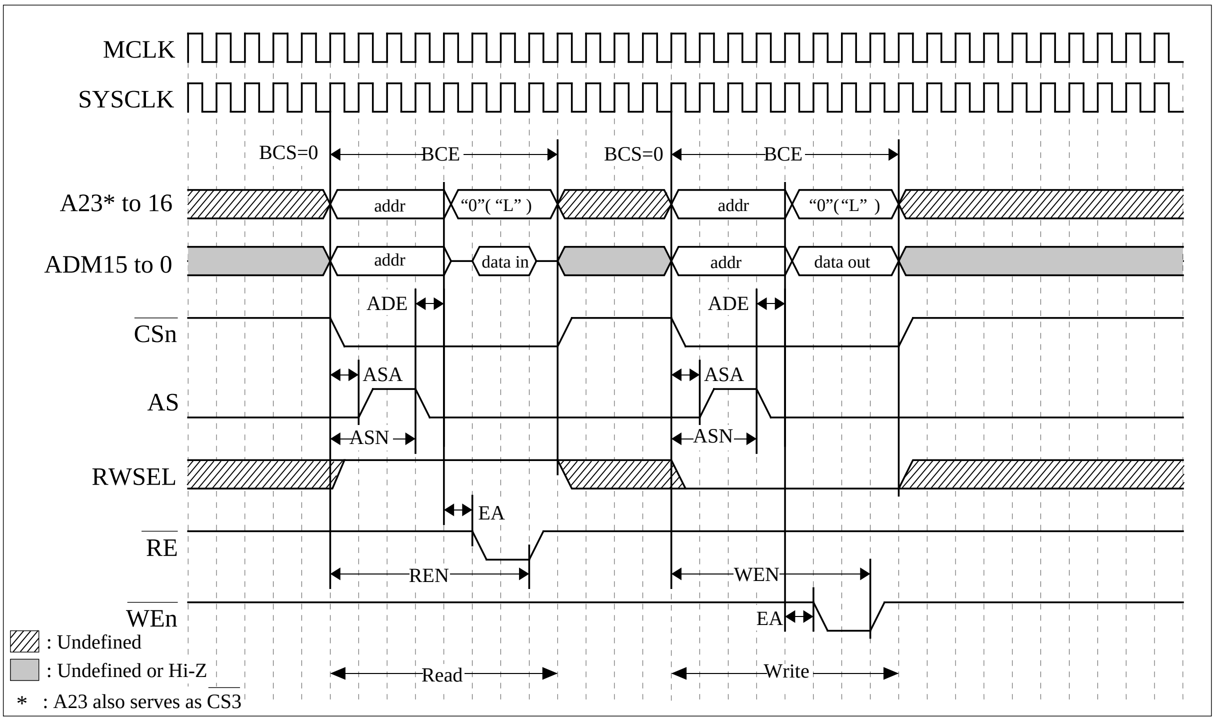


Fig. 8-13-19 Access Timing on a 16-bit Bus with Fixed Wait States, in Synchronous Mode and in Address/Data Multiplex Mode (MCLK = SYSCLK)

For details on the various timing settings, refer to the description of the memory control register in section 8.6, "Description of Registers."

8.13.8 16-bit Bus with Handshaking, in Synchronous Mode and in Address/Data Multiplex Mode

By setting the mode through the MMOD1 and 0 pins and the EXMOD1 and 0 pins, blocks 2 and 3 use multiplex pins for the memory address and memory data signals (pins ADM15 to 0).

When using handshaking, bus access starts once synchronization with SYSCLK is achieved, and after the data acknowledge signal ($\overline{\text{DK}}$) is asserted, 2 MCLK cycles are consumed by the BC internally and then the access is completed according to the specified parameters.

The various parameters for external memory access are set in memory control registers 2 and 3, corresponding to each block.

Fig. 8-13-20 is the timing chart in the case of a “16-bit bus with handshaking, in address/data multiplex mode, and with the frequency of MCLK equal to that of SYSCLK multiplied by four.”

Fig. 8-13-21 is the timing chart in the case of a “16-bit bus with handshaking, in address/data multiplex mode, and with the frequency of MCLK equal to that of SYSCLK multiplied by two.”

Fig. 8-13-22 is the timing chart in the case of a “16-bit bus with handshaking, in address/data multiplex mode, and with the frequency of MCLK equal to that of SYSCLK.”

As shown in each timing chart, the ADM15 to 0 pins go to “Hi-Z” or the undefined output state while $\overline{\text{CSn}}$ is negated in address/data multiplex mode. When the bus authority is released, the ADM15 to 0 pins are either pulled up or go to “Hi-Z”, depending on the setting of the I/O port output mode register.

The $\overline{\text{DK}}$ signal connected to the microcontroller should be input so as to be asserted from point EA+DW onward, and is negated before the next access.

Note that when writing to byte 0, $\overline{\text{WE0}}$ is asserted and the data is output on ADM7 to 0, and when writing to byte 1, $\overline{\text{WE1}}$ is asserted and the data is output on ADM15 to 8.

In addition, in the case of a word access (32 bits), the external access is performed twice with A[1] = “0” and A[1] = “1”.

Note: For details on the mode settings, refer to Table 8-9-1, “Mode Settings by the BC External Pins.”

Note: If handshaking mode is set for memory block 3, the only settings that are permitted are those in which MCLK is equal to SYSCLK multiplied by 4. Any setting in which MCLK is only twice SYSCLK, or in which the two frequencies are equal, is prohibited.

Note: “0” (low level) is output on pins A23 to 16 (A23 also serves as $\overline{\text{CS3}}$) while the ADM15 to 0 pins function as data pins. Therefore, refer to 3. in section 8.16, “Cautions,” regarding the use of these pins.

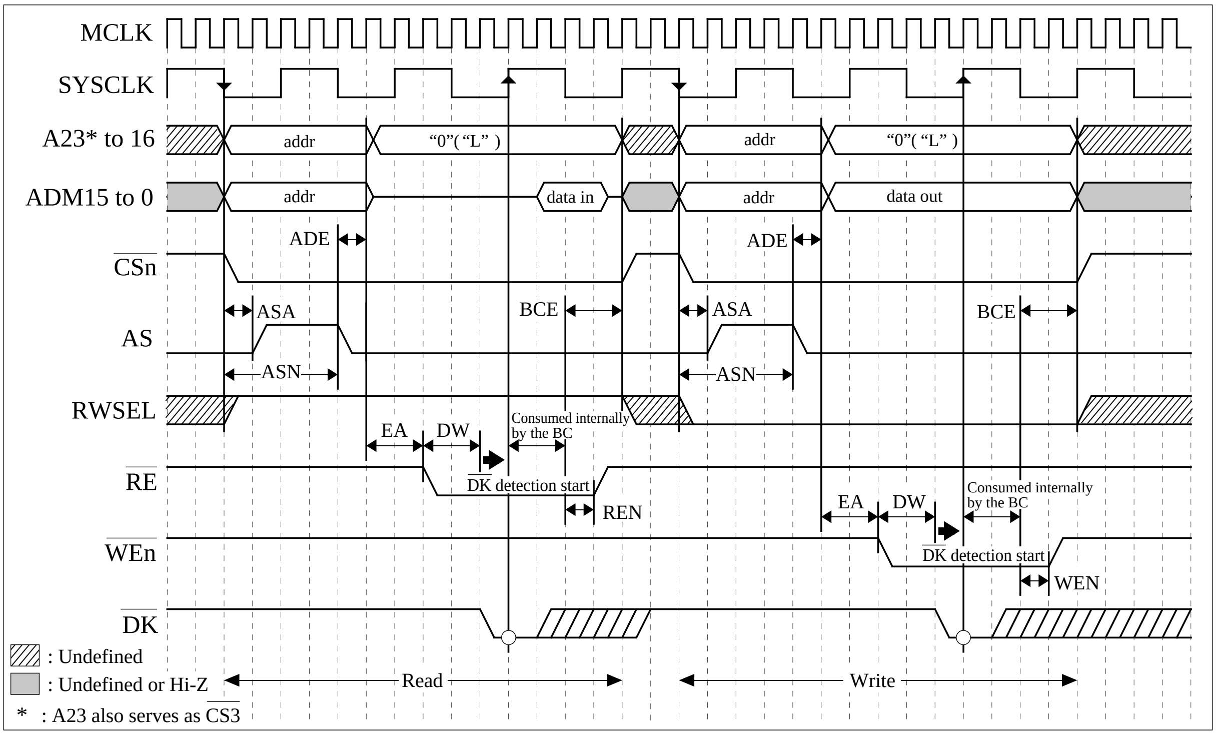


Fig. 8-13-20 Access Timing on a 16-bit Bus with Handshaking, in Synchronous Mode and in Address/Data Multiplex Mode (MCLK = SYSCLK multiplied by 4)

For details on the various timing settings, refer to the description of the memory control register in section 8.6, "Description of Registers."

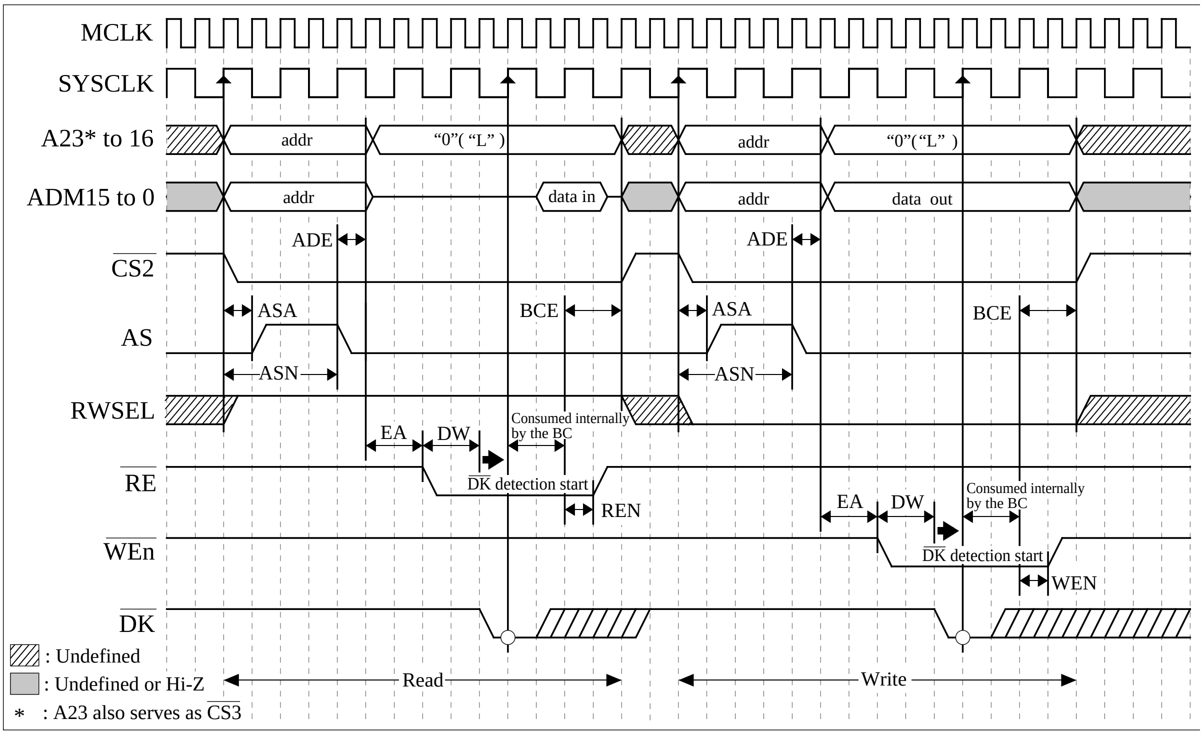


Fig. 8-13-21 Access Timing on a 16-bit Bus with Handshaking, in Synchronous Mode and in Address/Data Multiplex Mode (MCLK = SYSCLK multiplied by 2)

For details on the various timing settings, refer to the description of the memory control register in section 8.6, "Description of Registers."

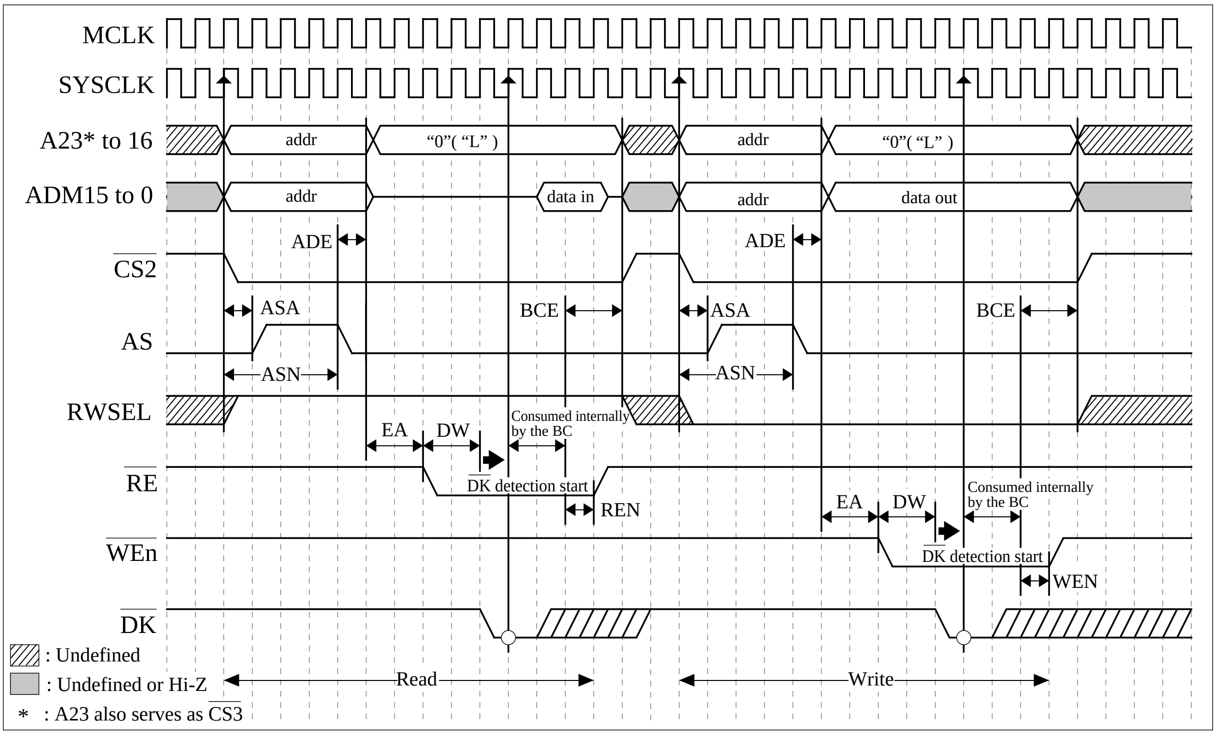


Fig. 8-13-22 Access Timing on a 16-bit Bus with Handshaking, in Synchronous Mode and in Address/Data Multiplex Mode (MCLK = SYSCLK)

For details on the various timing settings, refer to the description of the memory control register in section 8.6, "Description of Registers."

8.13.9 16-bit Bus in Asynchronous Mode and in Address/Data Multiplex Mode

By setting the mode through the MMOD1 and 0 pins and the EXMOD1 and 0 pins, blocks 0 to 3 use multiplex pins for the memory address and memory data signals (pins ADM15 to 0).
Asynchronous mode is used for accessing external memory at high speed; the address signals, $\overline{\text{CS}}$ signals, etc., are output asynchronously with SYSCLK but in synchronization with the internal MCLK. In asynchronous mode, accesses are all by fixed wait insertion.
The various parameters for external memory access are set in memory control registers 0 to 3, corresponding to each block.
Fig. 8-13-23 is the timing chart in the case of a “16-bit bus in asynchronous mode, in address/data multiplex mode.” As shown in the timing chart, the ADM15 to 0 pins go to “Hi-Z” or the undefined output state while $\overline{\text{CSn}}$ is negated in address/data multiplex mode. When the bus authority is released, the ADM15 to 0 pins are either pulled up or go to “Hi-Z”, depending on the setting of the I/O port output mode register.
Note that when writing to byte 0, $\overline{\text{WE0}}$ is asserted and the data is output on ADM7 to 0, and when writing to byte 1, $\overline{\text{WE1}}$ is asserted and the data is output on ADM15 to 8.
In addition, in the case of a word access (32 bits), the external access is performed twice with A[1] = “0” and A[1] = “1”.
Note: For details on the mode settings, refer to Table 8-9-1, “Mode Settings by the BC External Pins.”

Note: “0” (low level) is output on pins A23 to 16 (A23 also serves as $\overline{\text{CS3}}$) while the ADM15 to 0 pins function as data pins. Therefore, refer to 3. in section 8.16, “Cautions,” regarding the use of these pins.

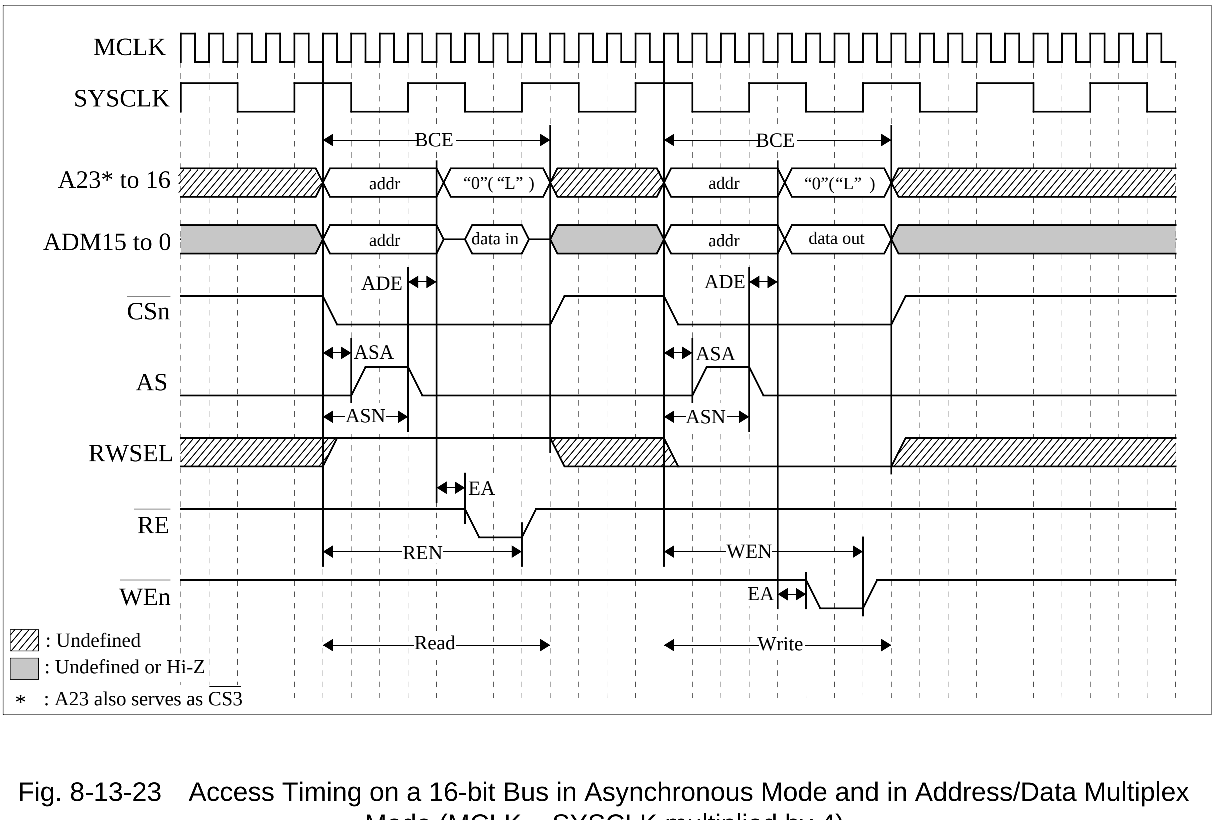


Fig. 8-13-23 Access Timing on a 16-bit Bus in Asynchronous Mode and in Address/Data Multiplex Mode (MCLK = SYSCLK multiplied by 4)

For details on the various timing settings, refer to the description of the memory control register in section 8.6, “Description of Registers.”

8.13.10 8-bit Bus with Fixed Wait States, in Synchronous Mode and in Address/Data Multiplex Mode

By setting the mode through the MMOD1 and 0 pins and the EXMOD1 and 0 pins, blocks 0 to 3 use multiplex pins for the memory address and memory data signals (pins ADM15 to 0).

8-bit bus mode is set for block 0 by setting the mode through the MMOD1 and 0 pins and the EXMOD1 and 0 pins, and for blocks 1 to 3 by setting the BnBW bit to “0” in the corresponding memory control register. In 8-bit bus mode, half-word access (16 bits) is performed by means of two external accesses, with A[0] = “0” for the low-order byte and A[0] = “1” for the high-order byte. Note that the low-order 8 bits (D7 to 0) are used for the data bus. Word access (32 bits) is performed by means of four accesses, with A[1:0] = “00”, A[1:0] = “01”, A[1:0] = “10”, and A[1:0] = “11”, starting from the low-order side.

In synchronous mode, the bus access starts in synchronization with SYSCLK, and when fixed wait states are inserted, the access ends according to the timing that was set in the memory control register.

The various parameters for external memory access are set in memory control registers 0 to 3, corresponding to each block.

Fig. 8-13-24 is the timing chart in the case of a half-word access using an “8-bit bus with fixed wait states, in synchronous mode, in address/data multiplex mode, and with the frequency of MCLK equal to that of SYSCLK multiplied by four.”

Fig. 8-13-25 is the timing chart in the case of a half-word access using an “8-bit bus with fixed wait states, in synchronous mode, in address/data multiplex mode, and with the frequency of MCLK equal to that of SYSCLK multiplied by two.”

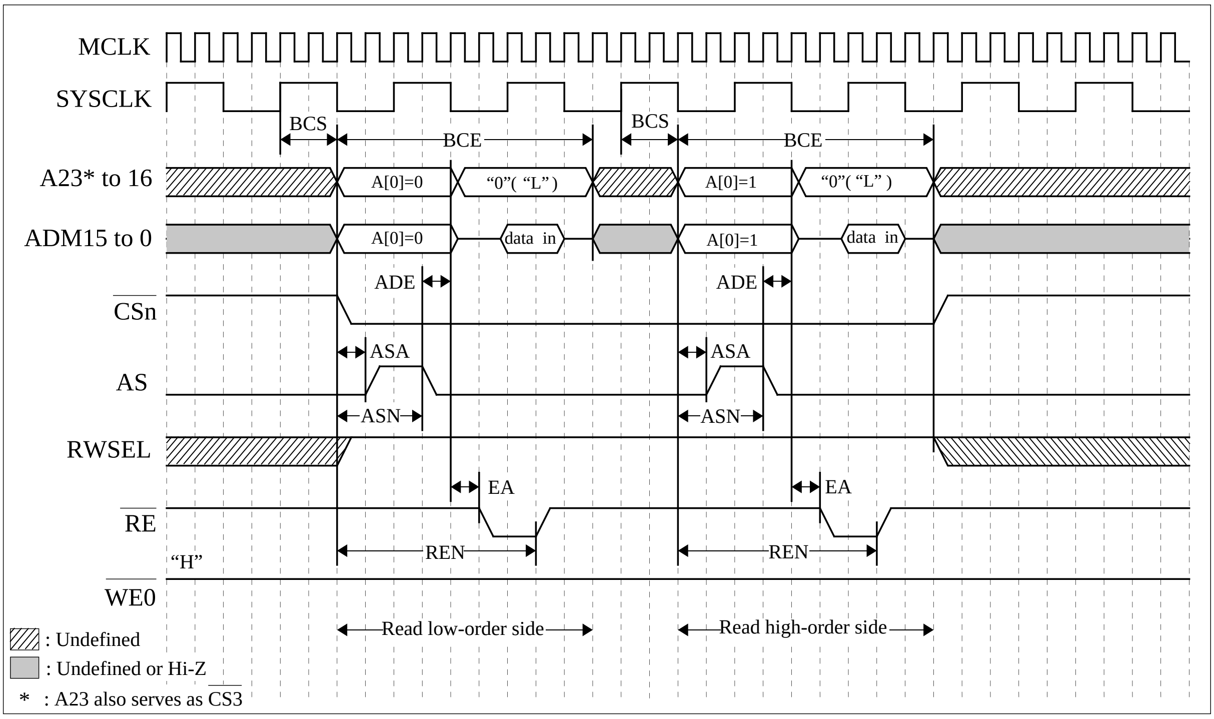
Fig. 8-13-26 is the timing chart in the case of a half-word access using an “8-bit bus with fixed wait states, in synchronous mode, in address/data multiplex mode, and with the frequency of MCLK equal to that of SYSCLK.”

As shown in each timing chart, the ADM15 to 0 pins go to “Hi-Z” or the undefined output state while $\overline{\text{CSn}}$ is negated in address/data multiplex mode. When the bus authority is released, the ADM15 to 0 pins are either pulled up or go to “Hi-Z”, depending on the setting of the I/O port output mode register.

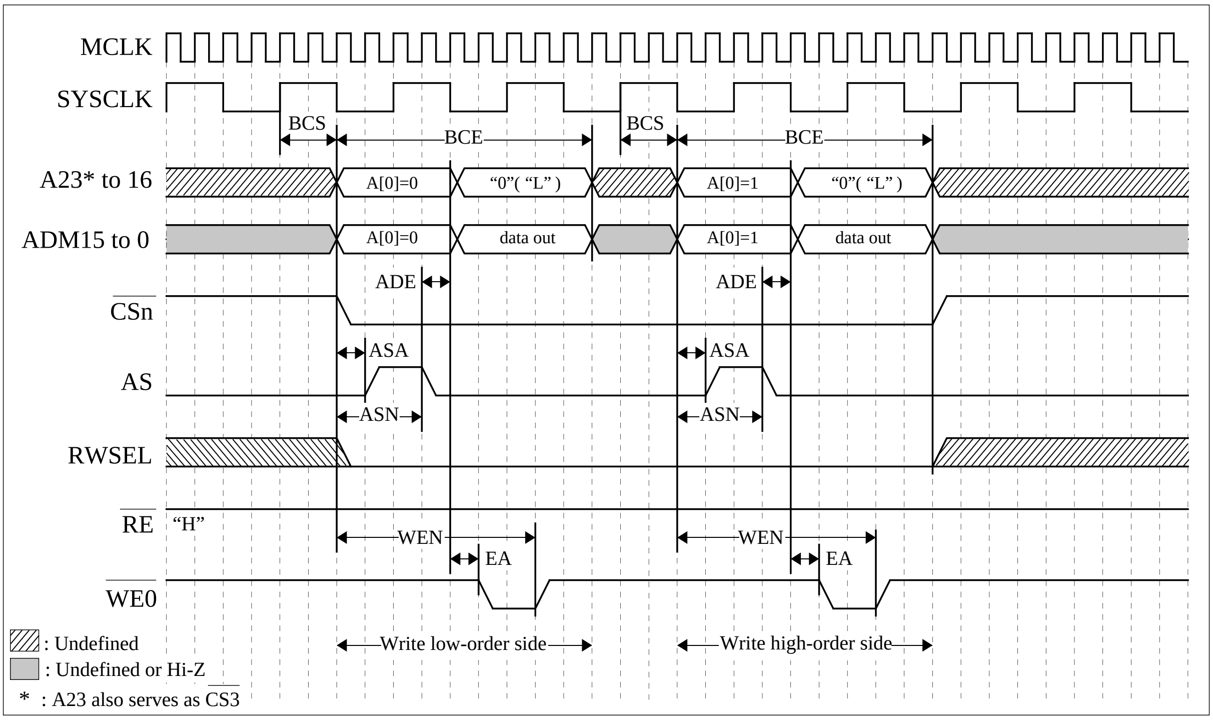
Note that when writing, $\overline{\text{WE0}}$ is asserted and the data is output on ADM7 to 0.

Note: For details on the mode settings, refer to Table 8-9-1, Mode Settings by the BC External Pins.”

Note: “0” (low level) is output on pins A23 to 16 (A23 also serves as $\overline{\text{CS3}}$) while the ADM15 to 0 pins function as data pins. Therefore, refer to 3. in section 8.16, “Cautions,” regarding the use of these pins.



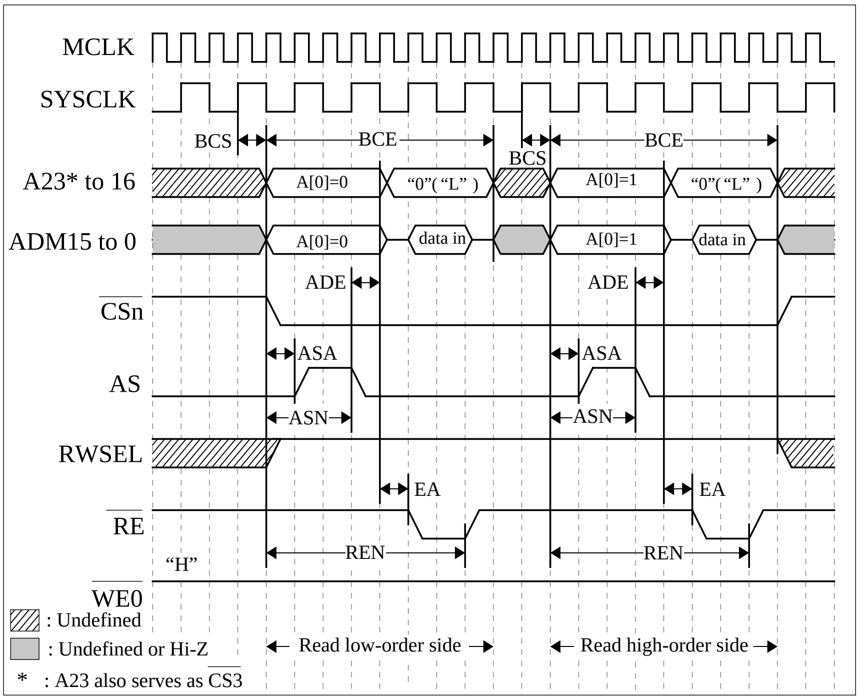
(a) Read Timing



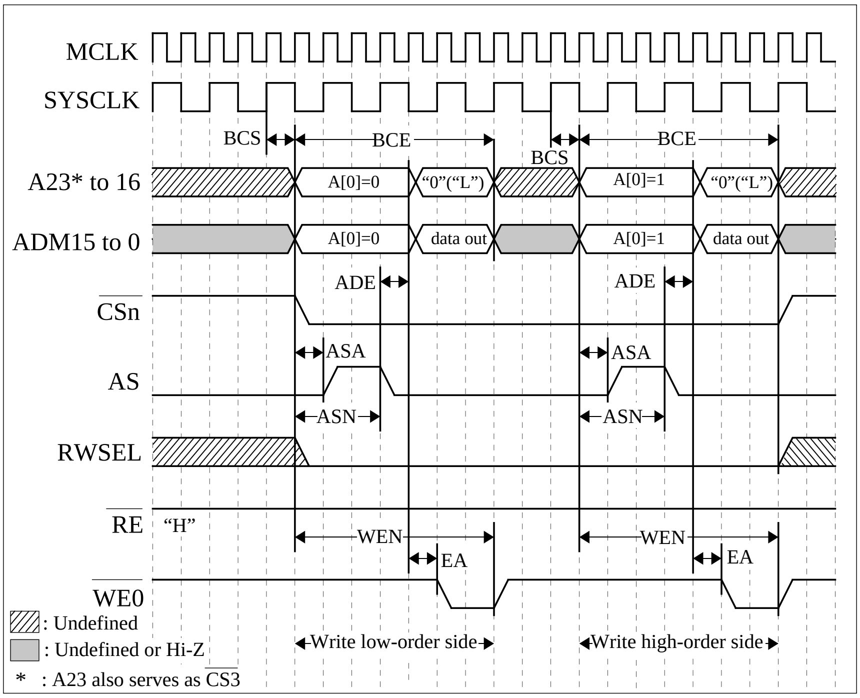
(b) Write Timing

Fig. 8-13-24 Access Timing on a 8-bit Bus with Fixed Wait States, in Synchronous Mode and in Address/Data Multiplex Mode (MCLK = SYSCLK multiplied by 4)

For details on the various timing settings, refer to the description of the memory control register in section 8.6, "Description of Registers."



(a) Read Timing

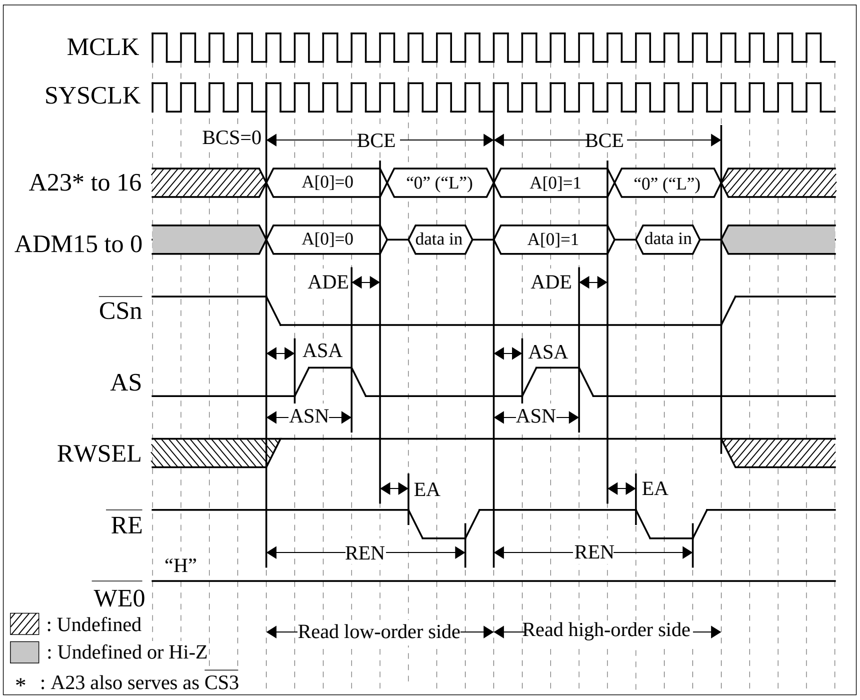


(b) Write Timing

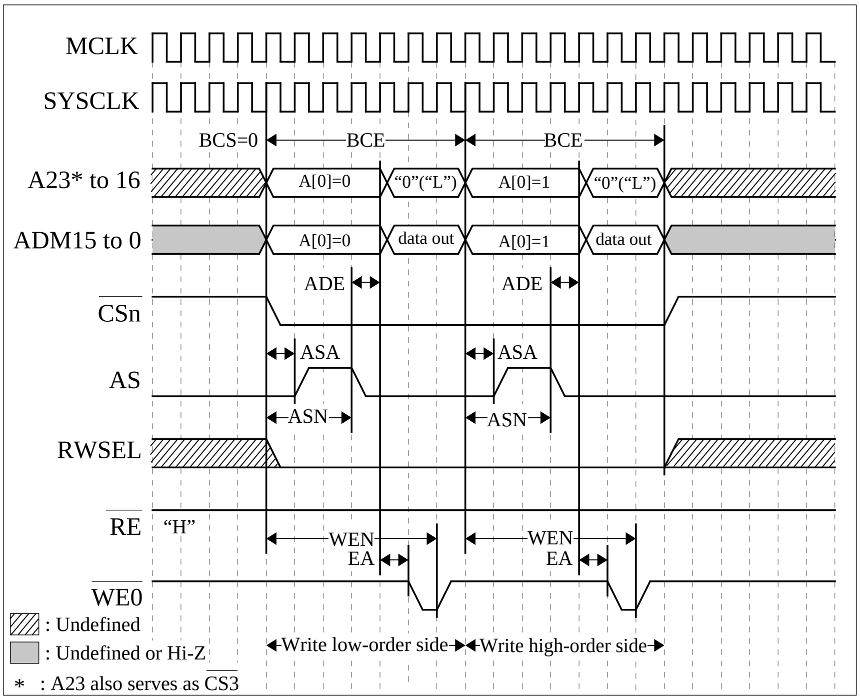
Fig. 8-13-25 Access Timing on a 8-bit Bus with Fixed Wait States, in Synchronous Mode and in Address/Data Multiplex Mode (MCLK = SYSCLK multiplied by 2)

For details on the various timing settings, refer to the description of the memory control register in section 8.6, "Description of Registers."

When BCE for the low-order bits is completed at the rising edge of SYSCLK, the bus cycle for the high-order bits begins at the same rising edge of SYSCLK.



(a) Read Timing



(b) Write Timing

Fig. 8-13-26 Access Timing on a 8-bit Bus with Fixed Wait States, in Synchronous Mode and in Address/Data Multiplex Mode (MCLK = SYSCLK)

For details on the various timing settings, refer to the description of the memory control register in section 8.6, "Description of Registers."

8.13.11 8-bit Bus with Handshaking, in Synchronous Mode and in Address/Data Multiplex Mode

By setting the mode through the MMOD1 and 0 pins and the EXMOD1 and 0 pins, blocks 2 and 3 use multiplex pins for the memory address and memory data signals (pins ADM15 to 0).

8-bit bus mode is set for blocks 2 and 3 by setting the BnBW bit to “0” in the corresponding memory control register.

In 8-bit bus mode, half-word access (16 bits) is performed by means of two external accesses, with A[0] = "0" for the low-order byte and A[0] = “1” for the high-order byte. Note that the low-order 8 bits (D7 to 0) are used for the data bus. Word access (32 bits) is performed by means of four accesses, with A[1:0] = “00”, A[1:0] = “01”, A[1:0] = “10”, and A[1:0] = “11”, starting from the low-order side.

With handshaking, bus access commences in synchronization with SYSCLK, and after the data acknowledge signal ($\overline{\text{DK}}$) is asserted, two MCLK cycles are consumed by the BC internally, and then access is completed as per the designated parameters.

The various parameters for external memory access are set in memory control registers 2 and 3, corresponding to each block.

Fig. 8-13-27 is the timing chart in the case of a half-word access using an “8-bit bus with handshaking, in address/data multiplex mode, and with the frequency of MCLK equal to that of SYSCLK multiplied by four.”

Fig. 8-13-28 is the timing chart in the case of a half-word access using an “8-bit bus with handshaking, in address/data multiplex mode, and with the frequency of MCLK equal to that of SYSCLK multiplied by two.”

Fig. 8-13-29 is the timing chart in the case of a half-word access using an “8-bit bus with handshaking, in address/data multiplex mode, and with the frequency of MCLK equal to that of SYSCLK.”

As shown in each timing chart, the ADM15 to 0 pins go to “Hi-Z” or the undefined output state while $\overline{\text{CSn}}$ is negated in address/data multiplex mode. When the bus authority is released, the ADM15 to 0 pins are either pulled up or go to “Hi-Z”, depending on the setting of the I/O port output mode register.

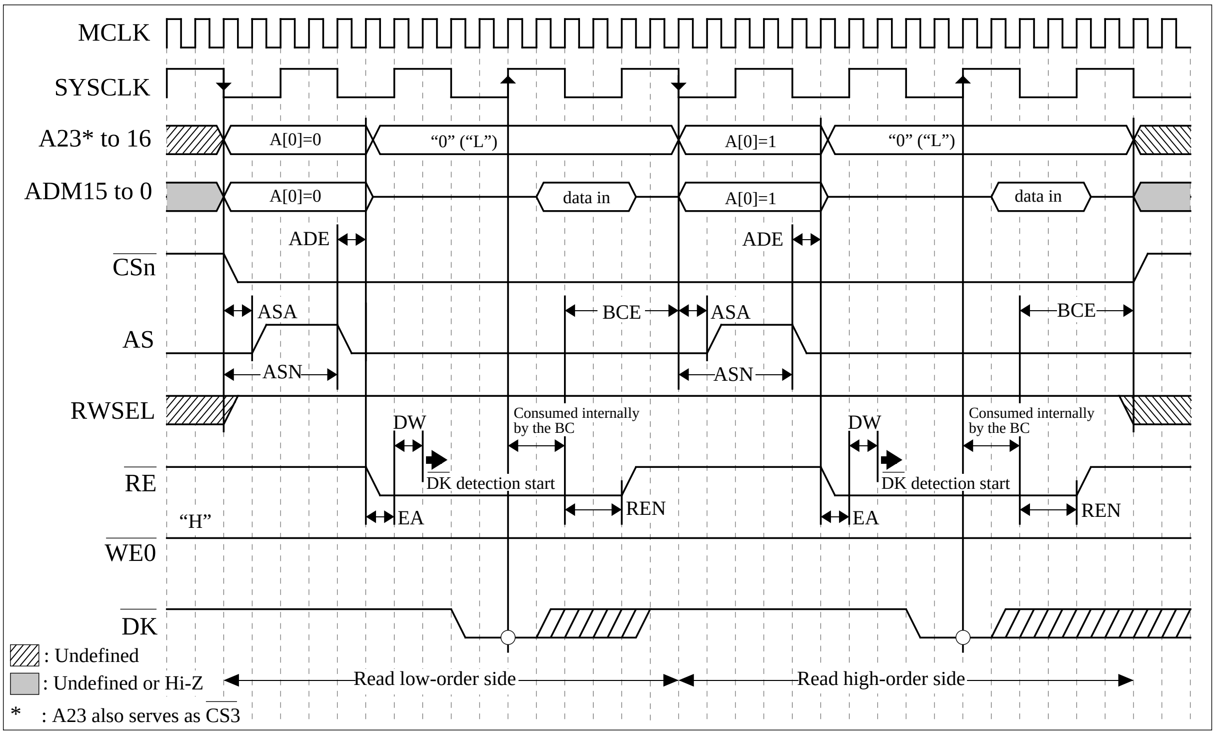
The $\overline{\text{DK}}$ signal connected to the microcontroller should be input so as to be asserted from point EA+DW onward, and is negated before the next access.

Note that when writing, $\overline{\text{WE0}}$ is asserted and the data is output on ADM7 to 0.

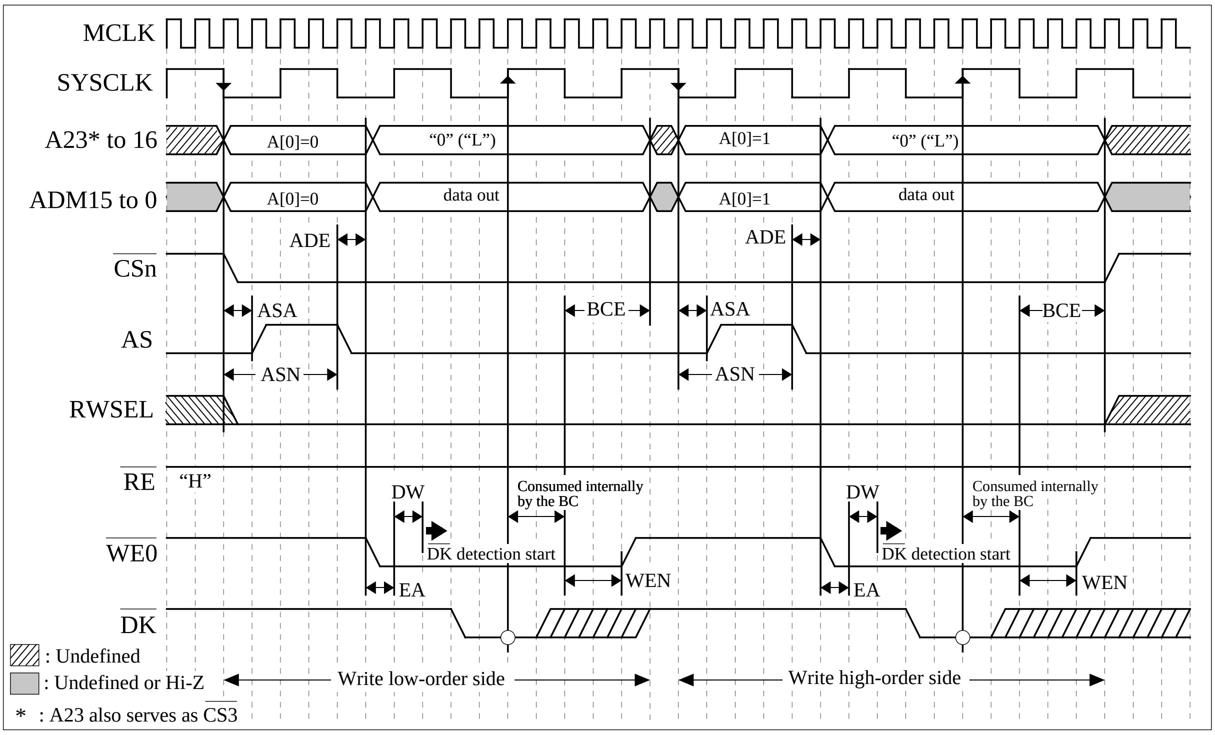
Note: For details on the mode settings, refer to Table 8-9-1, “Mode Settings by the BC External Pins.”

Note: If handshaking mode is set for memory block 3, the only settings that are permitted are those in which MCLK is equal to SYSCLK multiplied by 4. Any setting in which MCLK is only twice SYSCLK, or in which the two frequencies are equal, is prohibited.

Note: “0” (low level) is output on pins A23 to 16 (A23 also serves as $\overline{\text{CS3}}$) while the ADM15 to 0 pins function as data pins. Therefore, refer to 3. in section 8.16, “Cautions,” regarding the use of these pins.



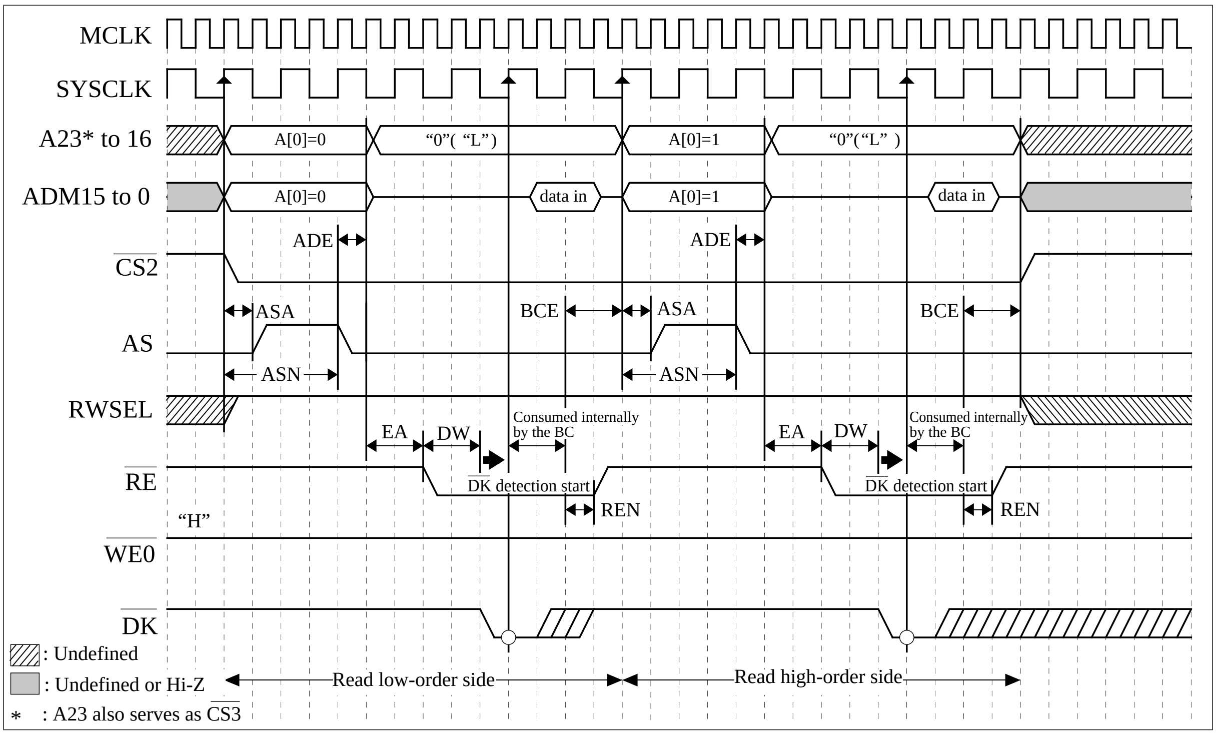
(a) Read Timing



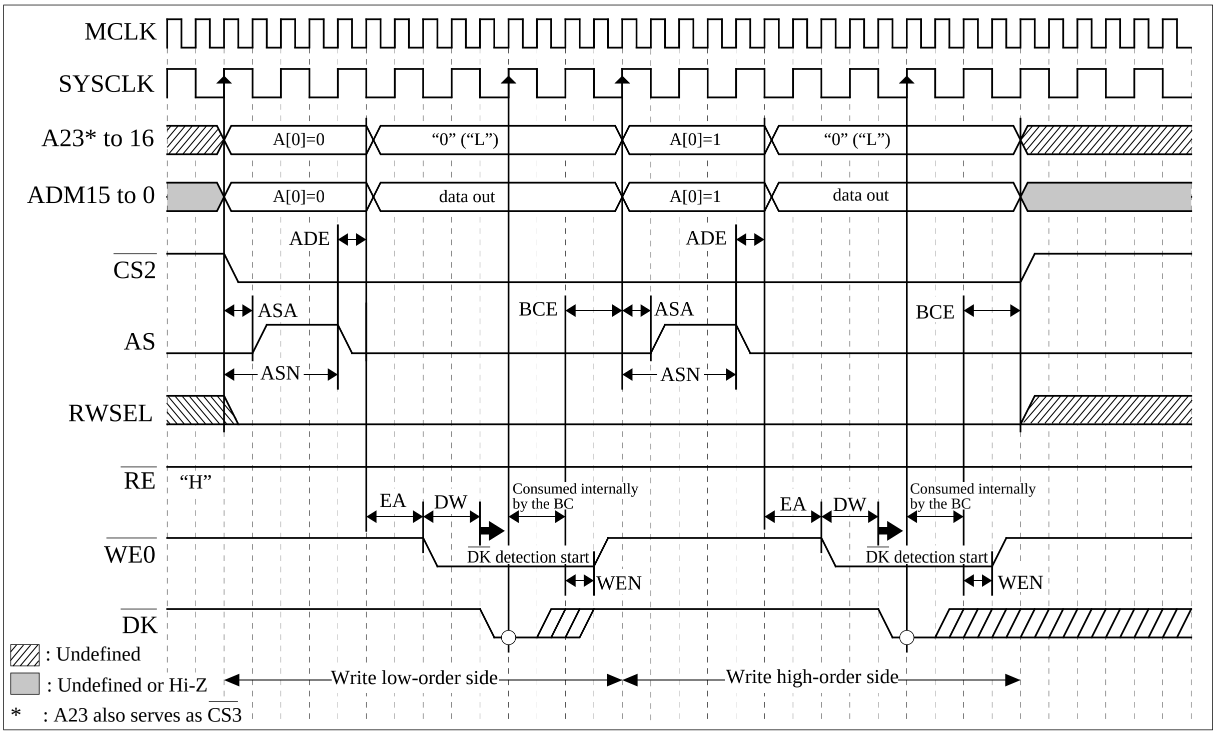
(b) Write Timing

Fig. 8-13-27 Access Timing on a 8-bit Bus with Handshaking, in Synchronous Mode and in Address/ Data Multiplex Mode (MCLK = SYSCLK multiplied by 4)

For details on the various timing settings, refer to the description of the memory control register in section 8.6, "Description of Registers."



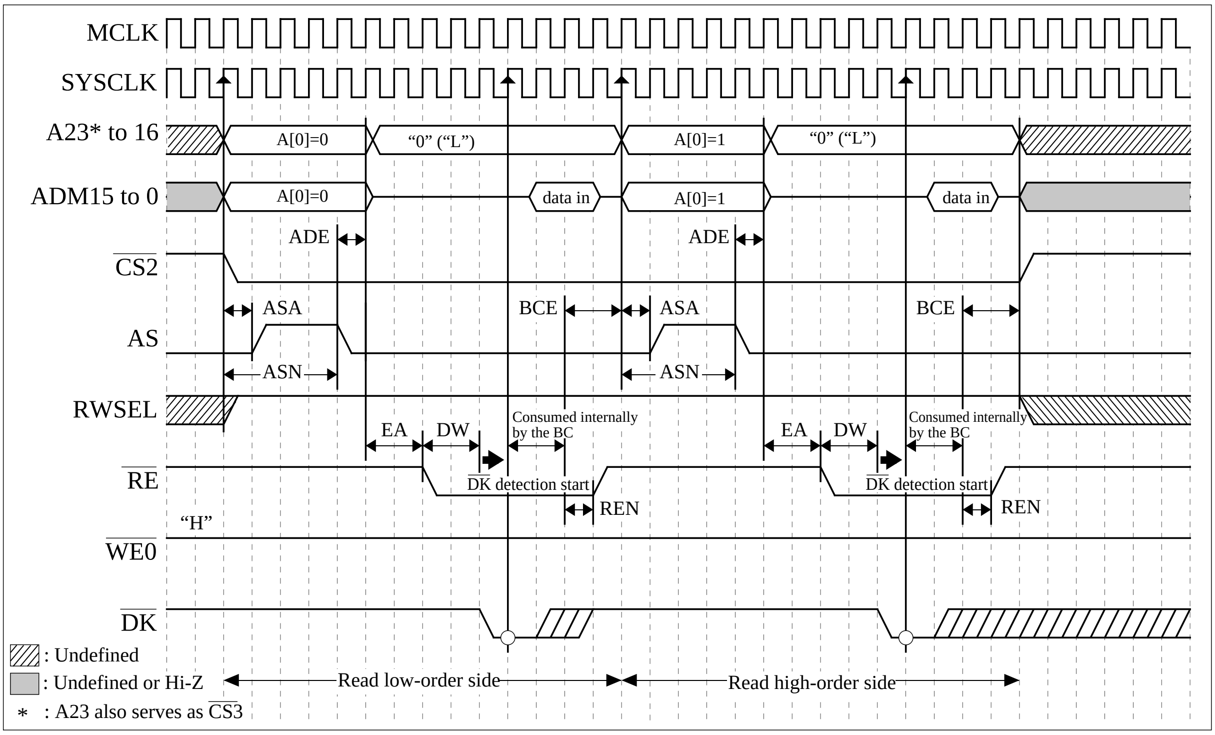
(a) Read Timing



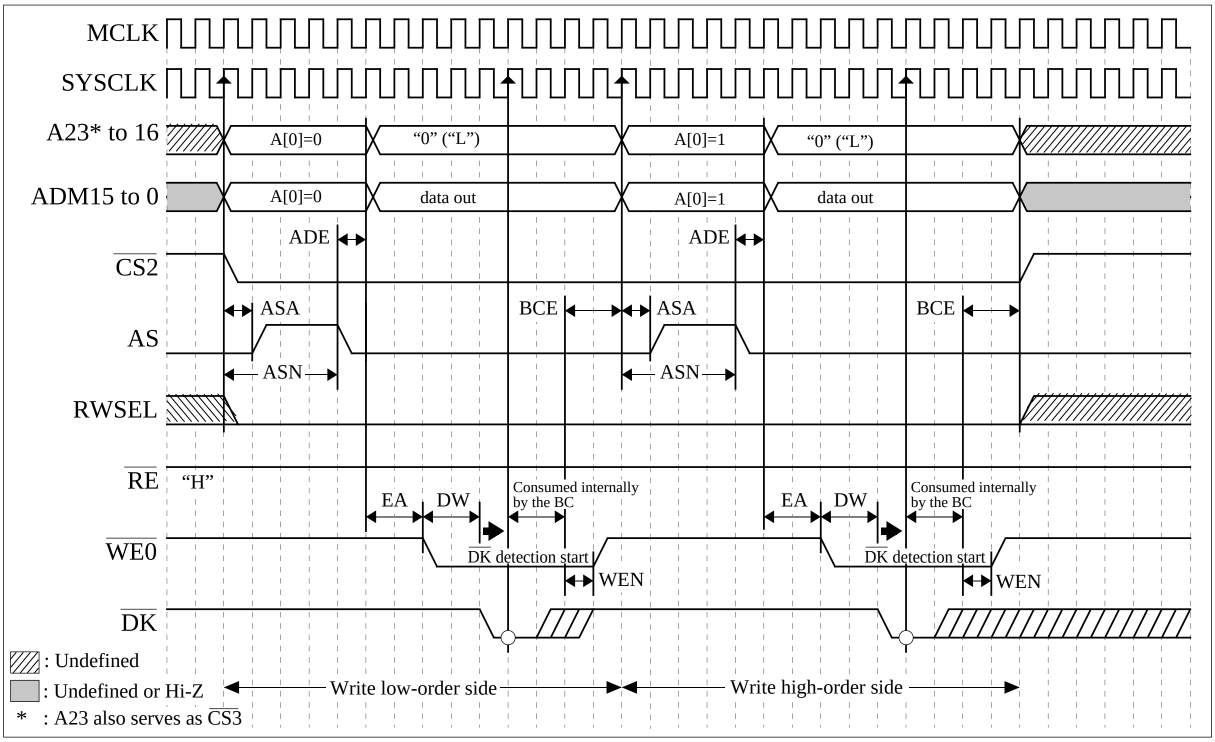
(b) Write Timing

Fig. 8-13-28 Access Timing on a 8-bit Bus with Handshaking in Synchronous Mode and in Address/ Data Multiplex Mode (MCLK = SYSCLK multiplied by 2)

For details on the various timing settings, refer to the description of the memory control register in section 8.6, "Description of Registers."



(a) Read Timing



(b) Write Timing

Fig. 8-13-29 Access Timing on a 8-bit Bus with Handshaking in Synchronous Mode and in Address/ Data Multiplex Mode (MCLK = SYSCLK)

For details on the various timing settings, refer to the description of the memory control register in section 8.6, "Description of Registers."

8.13.12 8-bit Bus in Asynchronous Mode and in Address/Data Multiplex Mode

By setting the mode through the MMOD1 and 0 pins and the EXMOD1 and 0 pins, blocks 0 to 3 use multiplex pins for the memory address and memory data signals (pins ADM15 to 0).

8-bit bus mode is set for block 0 by setting the mode through the MMOD1 and 0 pins and the EXMOD1 and 0 pins, and for blocks 1 to 3 by setting the BnBW bit to "0" in the corresponding memory control register. In 8-bit bus mode, half-word access (16 bits) is performed by means of two external accesses, with A[0] = "0" for the low-order byte and A[0] = "1" for the high-order byte. Word access (32 bits) is performed by means of four accesses, with A[1:0] = "00", A[1:0] = "01", A[1:0] = "10", and A[1:0] = "11", starting from the low-order side. Note that the low-order 8 bits (D7 to 0) are used for the data bus.

Asynchronous mode is used for accessing external memory at high speed; the address signals, $\overline{\text{CSn}}$ signals, etc., are output asynchronously with SYSCLK but in synchronization with the internal MCLK. In asynchronous mode, accesses are all by fixed wait insertion.

The various parameters for external memory access are set in memory control registers 0 to 3, corresponding to each block.

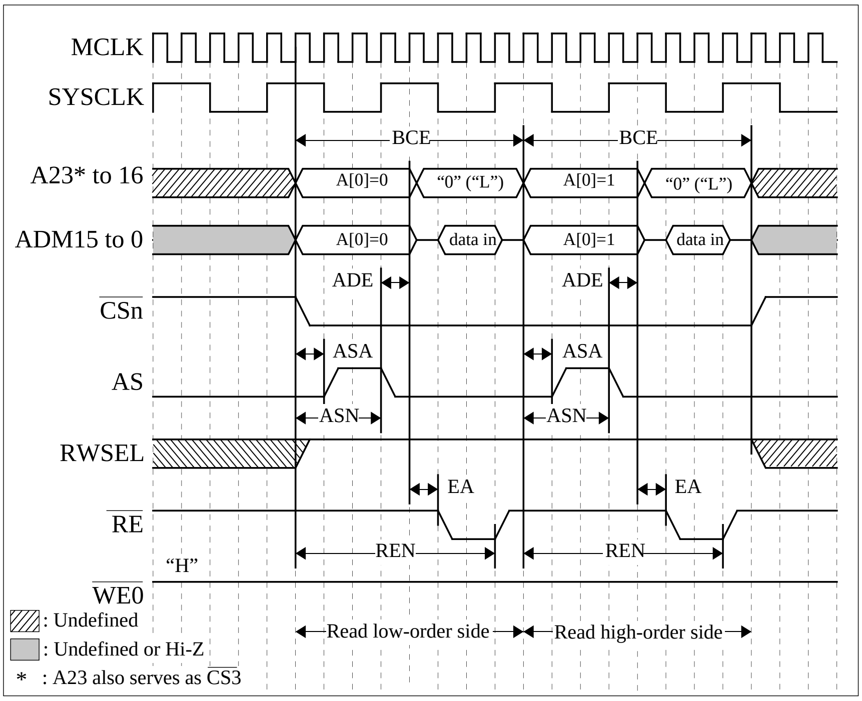
Fig. 8-13-30 is the timing chart in the case of a half-word access using an "8-bit bus in asynchronous mode, in address/data multiplex mode."

As shown in each timing chart, the ADM15 to 0 pins go to "Hi-Z" or the undefined output state while $\overline{\text{CSn}}$ is negated in address/data multiplex mode. When the bus authority is released, the ADM15 to 0 pins are either pulled up or go to "Hi-Z", depending on the setting of the I/O port output mode register.

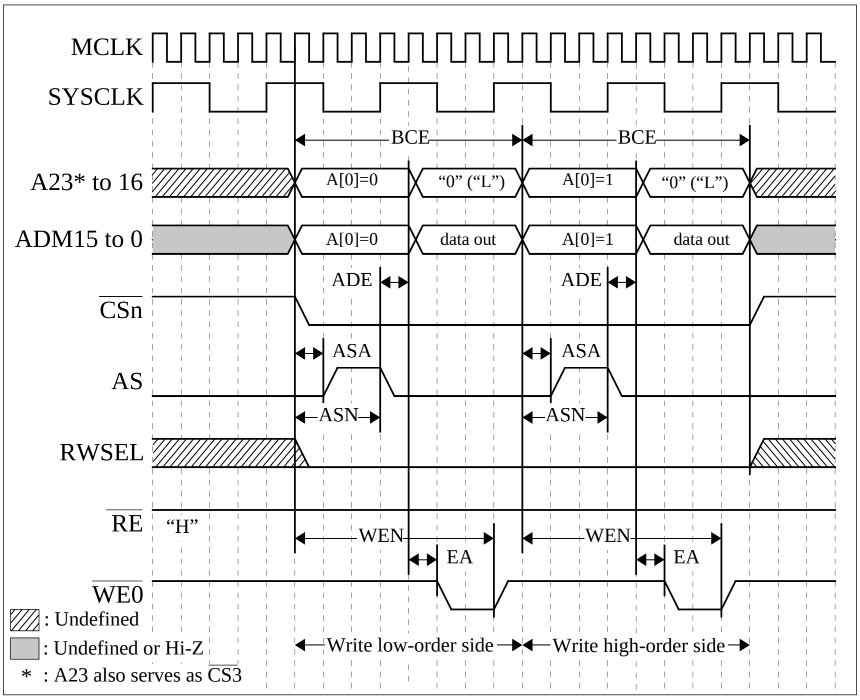
Note that when writing, $\overline{\text{WE0}}$ is asserted and the data is output on ADM7 to 0.

Note: For details on the mode settings, refer to Table 8-9-1, "Mode Settings by the BC External Pins."

Note: "0" (low level) is output on pins A23 to 16 (A23 also serves as $\overline{\text{CS3}}$) while the ADM15 to 0 pins function as data pins. Therefore, refer to 3. in section 8.16, "Cautions," regarding the use of these pins.



(a) Read Timing



(b) Write Timing

Fig. 8-13-30 Access Timing on a 8-bit Bus in Asynchronous Mode and in Address/Data Multiplex Mode (MCLK = SYSCLK multiplied by 4)

For details on the various timing settings, refer to the description of the memory control register in section 8.6, "Description of Registers."

8.14 External Memory Space Access (DRAM Space)

8.14.1 DRAM Space

Blocks 1 and 2 can be used as DRAM space by setting the BnDRAM bits in memory control registers 1B/2B and setting the DRAME bit in DRAM control register. The DRAM bus cycle is always not synchronized the external clock (but is synchronized with MCLK), and performs address multiplexed output, $\overline{\text{RAS}}$/$\overline{\text{CAS}}$ signal output, etc.

Note: When common pins are used for addresses and data, DRAM cannot be supported.

The $\overline{\text{RAS}}$/$\overline{\text{CAS}}$ signal output timing can be set through software in the DRAM control register and memory control registers 1A/B and 2A/B.
Fig. 8-14-1 shows the DRAM access timing chart.

Note: For details on the timing settings, refer to section 8.6, “Description of Registers.”

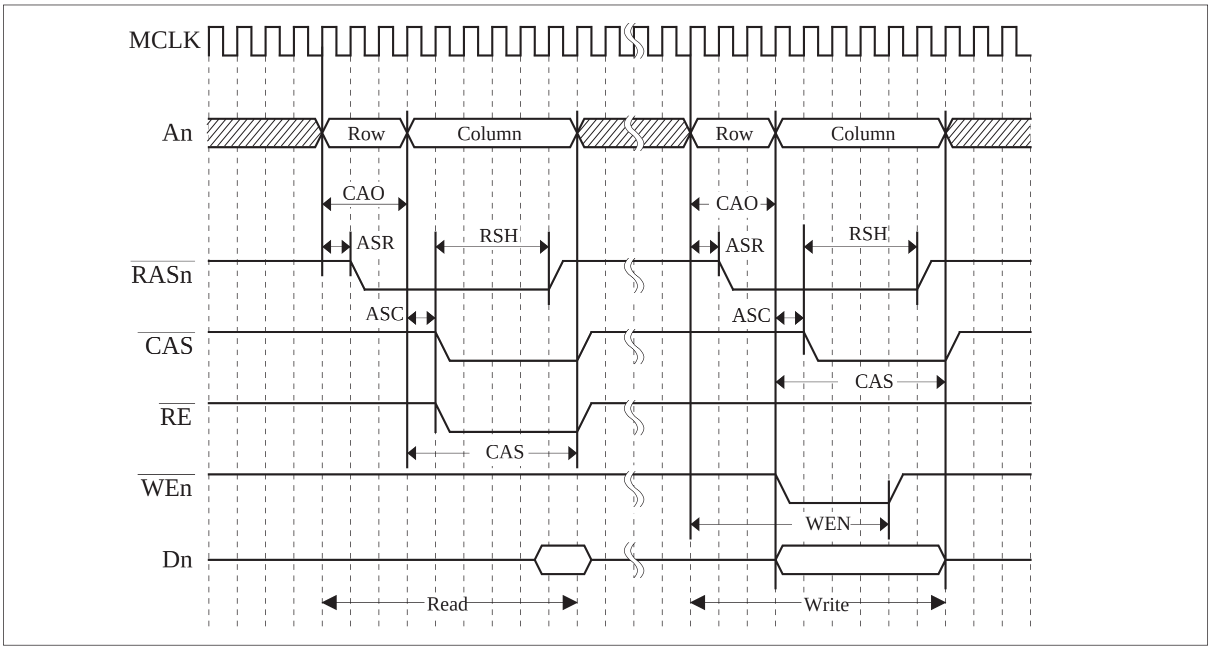


Fig. 8-14-1 DRAM Access Timing Chart

For details on the various timing settings, refer to the description of the memory control register in section 8.6, “Description of Registers.”

■ Minimum value for the $\overline{\text{RAS}}$ Precharge interval

When consecutive DRAM accesses are performed, the $\overline{\text{RAS}}$ precharge interval is shortest when performing an access of type (1) or (2) below while the PAGE bit is set to "0" in the DRAM control register:

(1) Word/half-word access while the bus width is set to 8 bits

(2) Word access while the bus width is set to 16 bits

Because the minimum value for the $\overline{\text{RAS}}$ precharge interval is:

RP + ASR

as shown in Fig. 8-14-2, set the parameters RP and ASR to values that will satisfy the DRAM requirements. Note that the minimum value that can be set for both RP and ASR is 1.

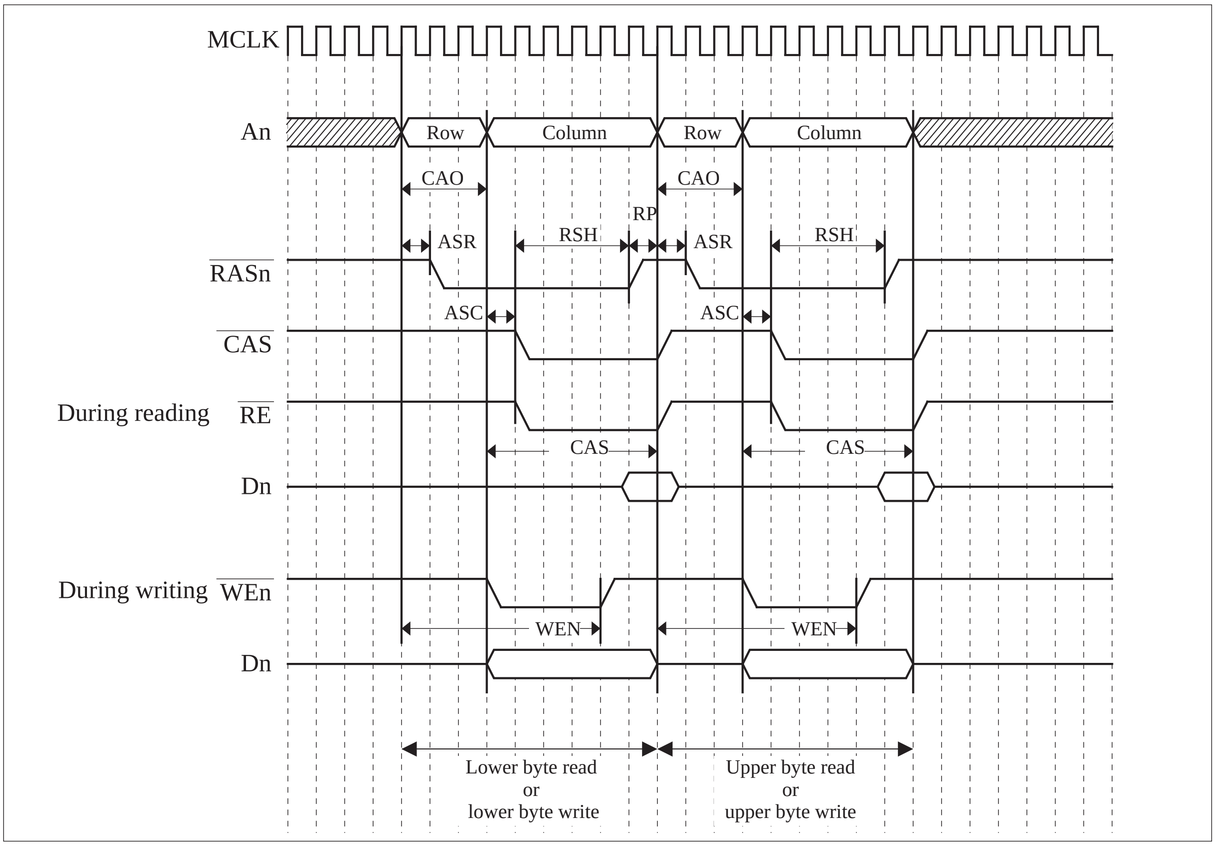


Fig. 8-14-2 Case Where the RAS Precharge Interval is at Its Minimum
(Example Where RP = 1 and ASR = 1)

■ 2 $\overline{\text{WE}}$ control/2 $\overline{\text{CAS}}$ control

DRAM that permits byte/word control can be supported by selecting either one of the following two methods:

- 2 $\overline{\text{WE}}$ control: The two pins $\overline{\text{WE1}}$ and $\overline{\text{WE0}}$ are used for byte/word control.
- 2 $\overline{\text{CAS}}$ control: The two pins $\overline{\text{DCAS1}}$ and $\overline{\text{DCAS0}}$ are used for byte/word control.

Fig. 8-14-3 illustrates an example of a write using 2 $\overline{\text{WE}}$ control, and Fig. 8-14-4 illustrates an example of a write using 2 $\overline{\text{CAS}}$ control.

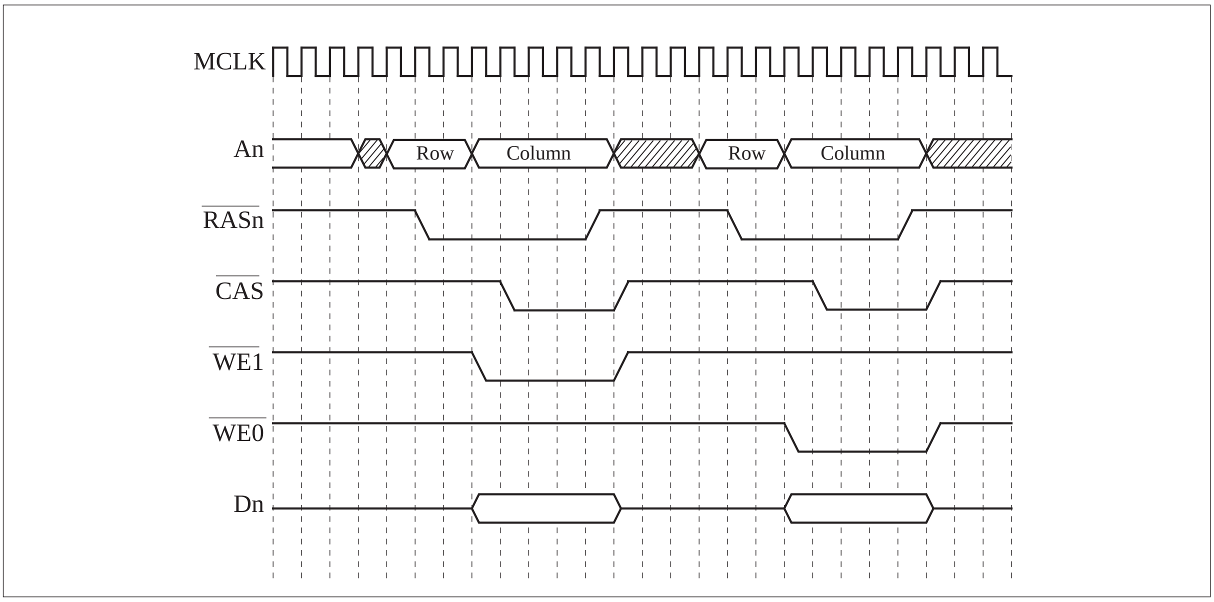


Fig. 8-14-3 Example of an 8-bit Data Write Using 2 $\overline{\text{WE}}$ Control (16-bit Bus Width)

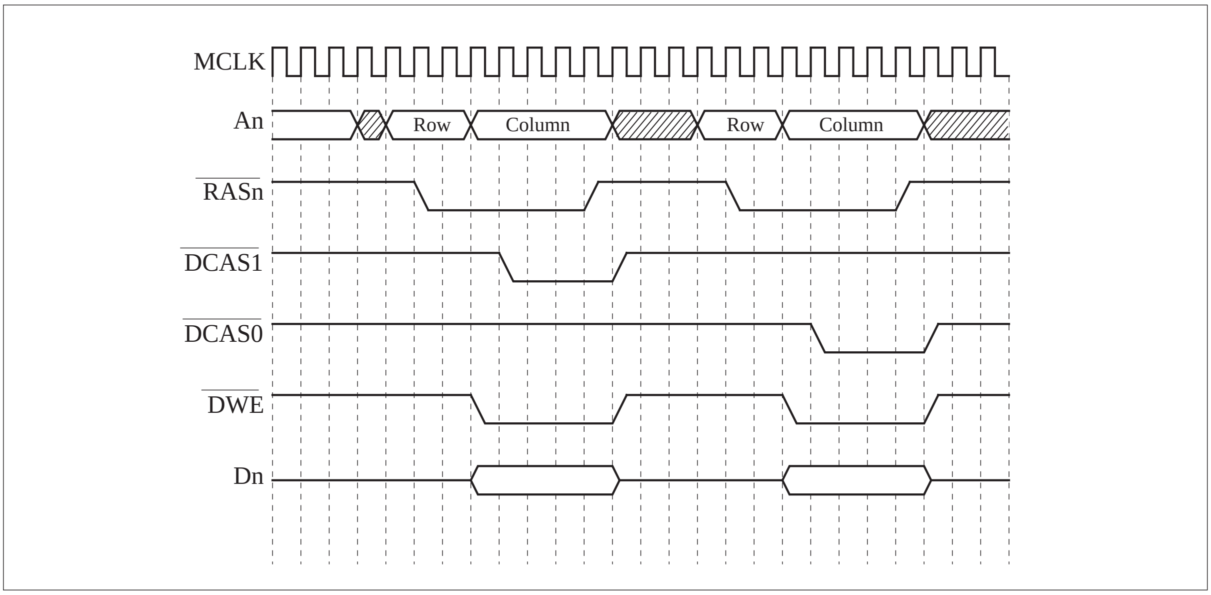


Fig. 8-14-4 Example of an 8-bit Data Write Using 2 $\overline{\text{CAS}}$ Control (16-bit Bus Width)

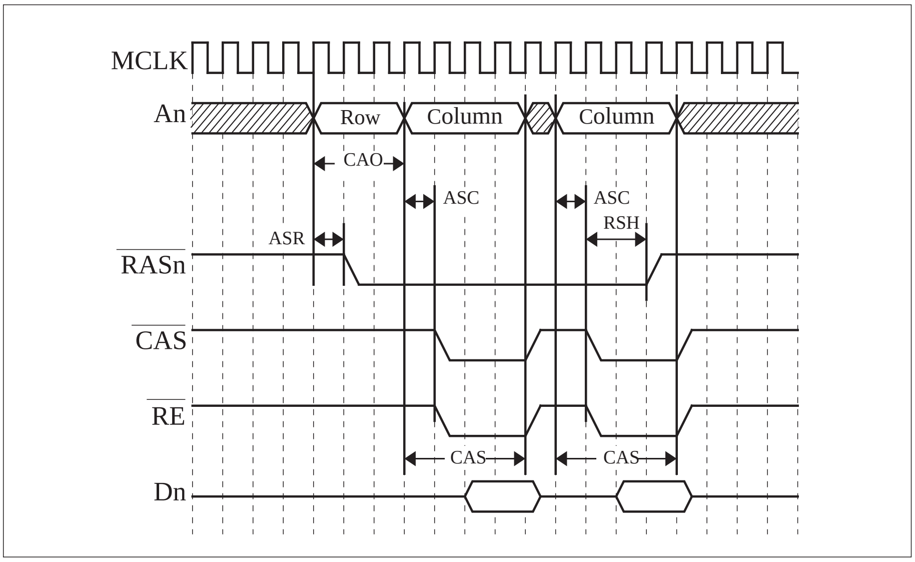
8.14.2 DRAM page mode

If the PAGE bit in the DRAM control register is set to “1”, page mode access is enabled, making high-speed access in page mode possible for following accesses to DRAM.

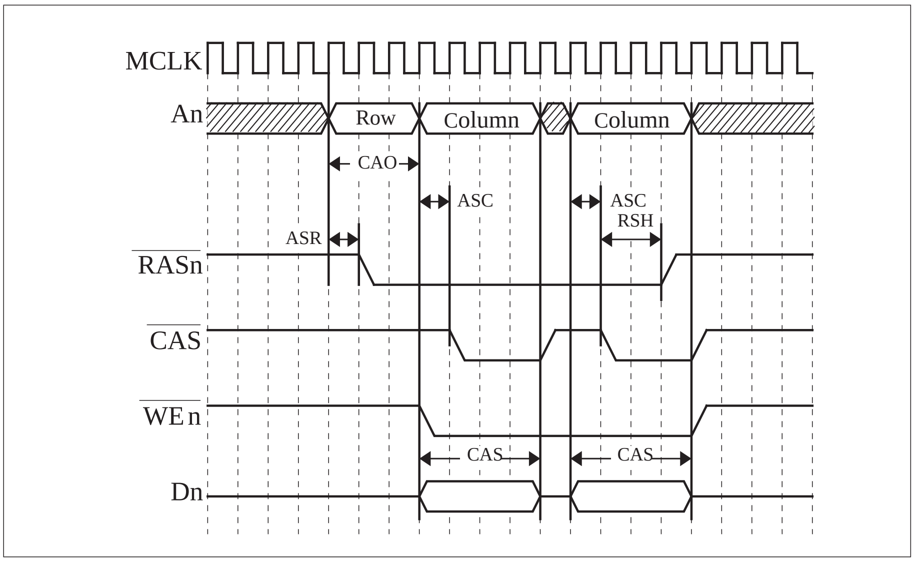
(1) Word/half-word access when the bus width is set to 8 bits

(2) Word access when the bus width is set to 16 bits

Fig. 8-14-5 shows the page mode read timing and write timing.



(a) Read Timing



(b) Write Timing

Fig. 8-14-5 DRAM Page Mode Read/Write Timing

For details on the various timing settings, refer to the description of the memory control register in section 8.6, “Description of Registers.”

8.14.3 Software Page Mode

Software page mode is a mode that forcibly initiates page mode by setting the control register.
Operation within software page mode is as described below. Refer to Fig. 8-14-6.

- When the mode is initiated, the contents of PRAR are output as the row address.
- While the mode is in effect, $\overline{\text{RASn}}$ for the block corresponding to the memory control register that initiated the mode is maintained in the asserted state.
- After the mode is initiated, external accesses are all processed as column address accesses. In this case, the ASC and CAS parameters of the block in question are guaranteed. In addition, the $\overline{\text{RE}}$/$\overline{\text{WE}}$ signal has the same waveform as in a normal DRAM access.
- The $\overline{\text{CAS}}$ precharge interval depends on the timing of the external access. Note that the shortest $\overline{\text{CAS}}$ precharge interval is ASC + 3 for a read, or ASC + 1 for a write. As an example, Fig. 8-14-6 illustrates the case where the $\overline{\text{CAS}}$ precharge interval is at its shortest.
- (CAO + 1) MCLK pulses are guaranteed for the row address output interval. ("CAO" is a parameter.)
- The parameter ASR for the block in question is guaranteed for the assertion of $\overline{\text{RASn}}$.

The procedure for executing this mode is described below:

■ Preparation for mode initiation

(1) Set the row address in the row address register PRAR.
Set a row address that has already been shifted according to the DRAM size.

(2) Set DRAMCTR.
Be certain to set the PAGE bit to "1" and the DRAME bit to "1".

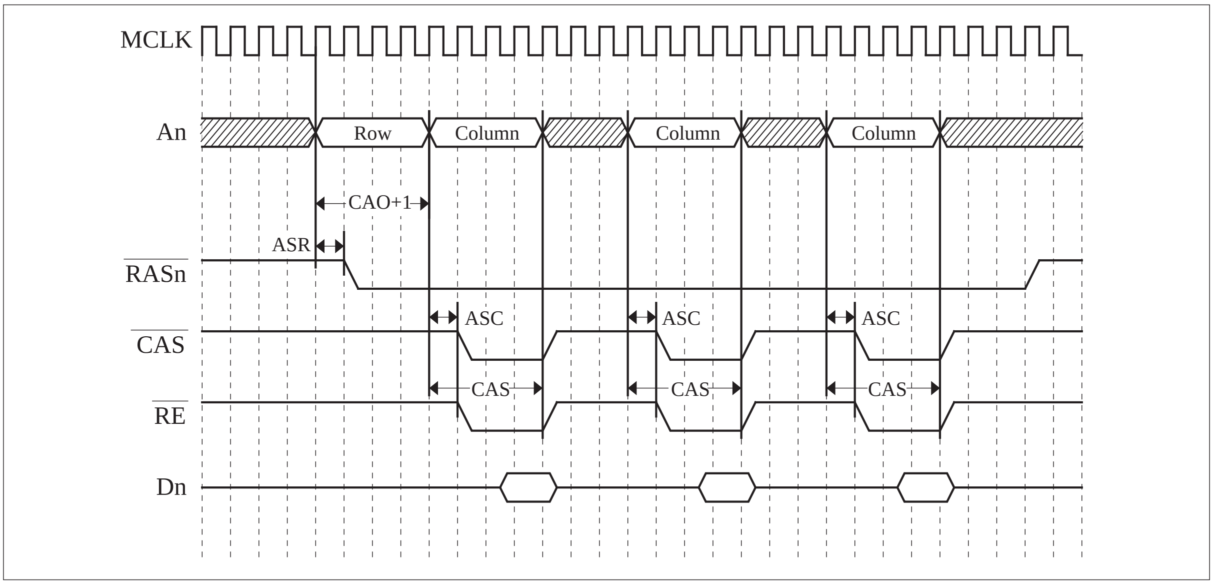
■ Mode initiation

Once the PE bit of the memory control register for the block in question (block 1 or 2) is set to "1", software page mode access begins after the register writing operation has been completed.

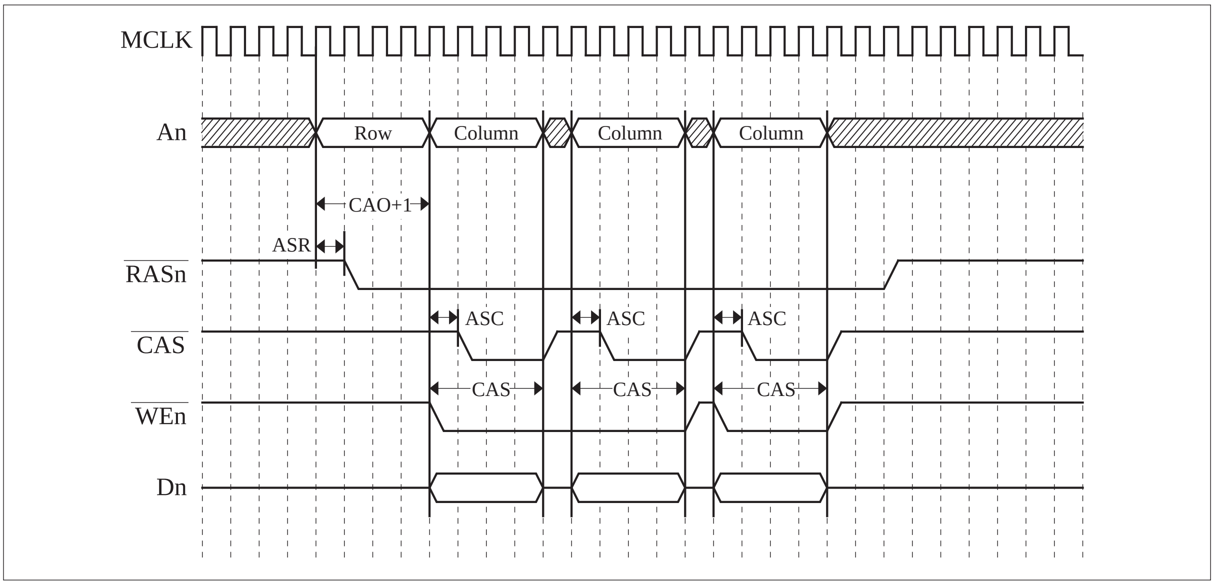
■ Mode termination

Once the PE bit of the memory control register for the block in question (block 1 or 2) is set to "0", software page mode is terminated after the register writing operation has been completed.

Note: When performing ICE trace/emulation in software page mode, set the CAS parameter to "5" or higher.



(a) Read Timing



(b) Write Timing

Fig. 8-14-6 Software Page Mode Read/Write Timing

For details on the various timing settings, refer to the description of the memory control register in section 8.6, “Description of Registers.”

[Restrictions on Use]

(1) While software page mode is in effect, external access outside of the block in question is prohibited.
Cancel software page mode before accessing an external memory space other than the block for which software page mode is set.

(2) While software page mode is in effect, the bus will not be released, regardless of any accesses to DRAM, even if the bus request signal $\overline{\text{BR}}$ is asserted.
If it is necessary to accept the bus request signal $\overline{\text{BR}}$ while data is being transferred in software page mode, partition the volume of data that is to be transferred in software page mode. Then, the program should temporarily release software page mode once after the transfer of each block of data is completed. If the bus request signal $\overline{\text{BR}}$ is being asserted when the software page mode is released, the bus grant signal $\overline{\text{BG}}$ is asserted, and the bus is released.

(3) The DRAM refresh cycle is not performed while software page mode is in effect.
If DRAM refresh is necessary, temporarily release DRAM software page mode once within each refresh cycle.

(4) While software page mode is in effect, any access that writes a “1” to the PE bit of the memory control register that initiated the mode is prohibited. ($\overline{\text{RASn}}$ remains asserted, and software page mode starts over again from the output of the row address.)

8.14.4 DRAM refresh

If the REFE bit in the DRAM control register is set, CAS-before-RAS refresh is performed at the interval set by the refresh count register. Fig. 8-14-7 illustrates the refresh operation concept. The refresh interval is the product of the value of the REFCNT in the refresh count register. If the REFE bit in the DRAM control register is set, the refresh count register operates as a down-counter, and the refresh count value is counted from REFC, the value of REFCNT, to 0. The refresh operation is executed once in an idle external bus cycle during the period while the refresh count value is counted down from REFC to 0. If, due to a serial interface access or other such operation, there is no idle external bus cycle before the refresh count value reaches 0, then a refresh cycle is inserted right after the bus cycle that is being executed at the moment the refresh count value reaches 0 is completed.
Fig. 8-14-8 shows the timing of a CAS-before-RAS refresh operation.

For details on the DRAM refresh interval setting, refer to section 8.6.6, "Refresh Count Register."

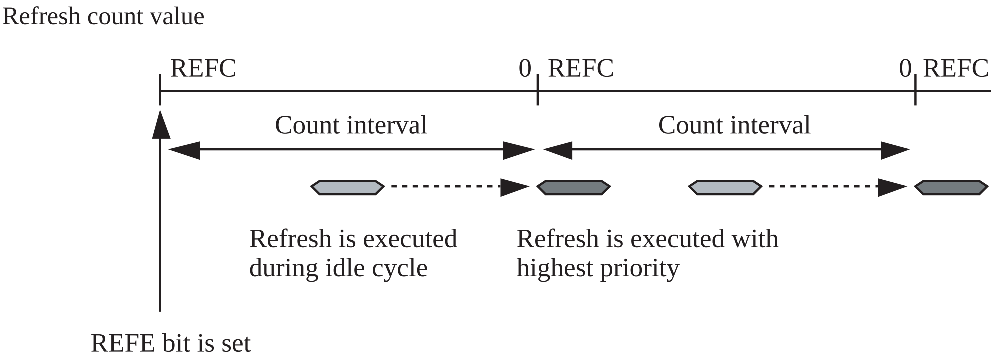

Refresh count value
REFC
0
REFC
0
REFC
Count interval
Count interval
Refresh is executed during idle cycle
Refresh is executed with highest priority
REFE bit is set

Fig. 8-14-7 DRAM Refresh Operation

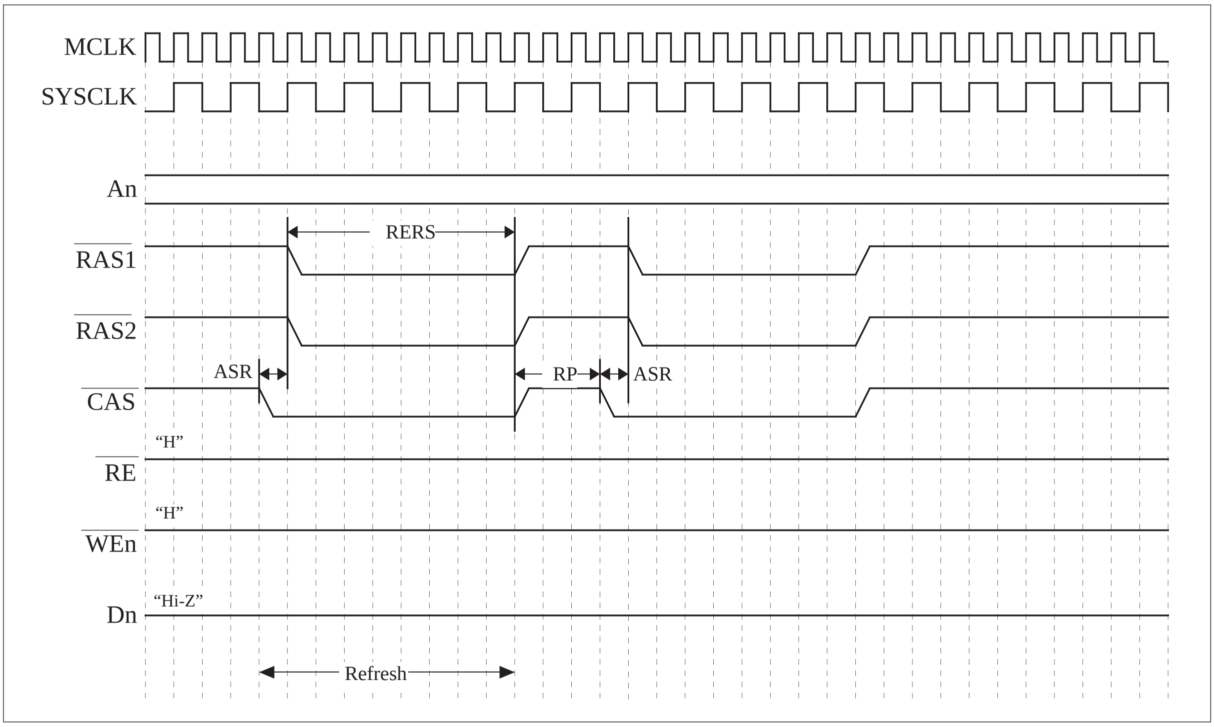


Fig. 8-14-8 DRAM Refresh Timing

For details on the ASR and RP settings, refer to the explanations in section 8.6.2, “Memory Block 1 Control Register,” and section 8.6.3, “Memory Block 2 Control Register.”
For details on the RERS setting, refer to the explanation in section 8.6.5, “DRAM Control Register.”

Note: When using blocks 1 and 2 as DRAM space simultaneously, the timing (ASR, RP) set in memory control register 1A/B is used as the refresh timing for both block 1 and block 2.

8.15 Bus Arbitration

In this microcontroller, bus arbitration is implemented through the bus authority request signal ($\overline{\text{BR}}$) and the bus authority release signal ($\overline{\text{BG}}$).

If an external device asserts the $\overline{\text{BR}}$ signal, then once the current bus access that is being executed is completed, the $\overline{\text{BG}}$ signal is asserted and the bus authority is released to the external device. Once the $\overline{\text{BR}}$ signal is negated, this LSI negates the $\overline{\text{BG}}$ signal in order to re-acquire the bus authority. However, if a refresh request is generated by the DRAM control circuit within this microcontroller while the bus authority has been released to an external device, this LSI negates the $\overline{\text{BG}}$ signal and requests the bus authority back form the external device. The external device then negates the $\overline{\text{BR}}$ signal in response, and the refresh is executed.

Note that bus arbitration is performed in synchronization with SYSCLK.

Fig. 8-15-1 to 3 show the timing for releasing the bus authority to an external device, and Fig. 8-15-4 shows the timing when a refresh request is generated while the bus authority has been released. An, $\overline{\text{CSn}}$, $\overline{\text{RE}}$, $\overline{\text{WEn}}$, $\overline{\text{RASn}}$, and $\overline{\text{CAS}}$ (and, if there is output on other pins related to the BC, those signals as well) are always output by this microcontroller which has the bus authority ($\overline{\text{BG}}$ = “H”)*1, and go to “Hi-Z” (high impedance) when the bus authority is released ($\overline{\text{BG}}$ = “L”).

Note that the execution of internal I/O space access requests and external memory space access requests by the CPU while the bus authority is being released are delayed until the bus authority release is completed.

Accesses which can be executed while the bus authority is being released and accesses which are delayed until bus authority release is completed are listed below.

(1) Accesses which can be executed while the bus authority is being released
  - Internal data RAM space accesses by the CPU
  - Internal ROM/internal flash memory space accesses by the CPU

(2) Accesses which are delayed until bus authority release is completed
  - Internal I/O space accesses by the CPU
  - External memory space accesses by the CPU

Note: For details on pins related to the BC and their statuses, refer to Table 8-5-2, “Operating Status of Pins Concerning BC.”

*1) However, if a bus access is executed immediately before the bus authority is released to an external device, the An signal, etc., may go to high impedance, even if $\overline{\text{BG}}$ = “H”, depending on the timing of the completion of that bus access. Specifically, the An and other signals are placed in the high impedance state at the following timing relative to the timing at which the $\overline{\text{BG}}$ signal is asserted:

When nfr = 4: Simultaneously, or 1, 2, or 3 MCLK cycles before

When nfr = 2: Simultaneously, or 1 MCLK cycle before

Here, nfr = MCLK frequency/SYSCLK frequency

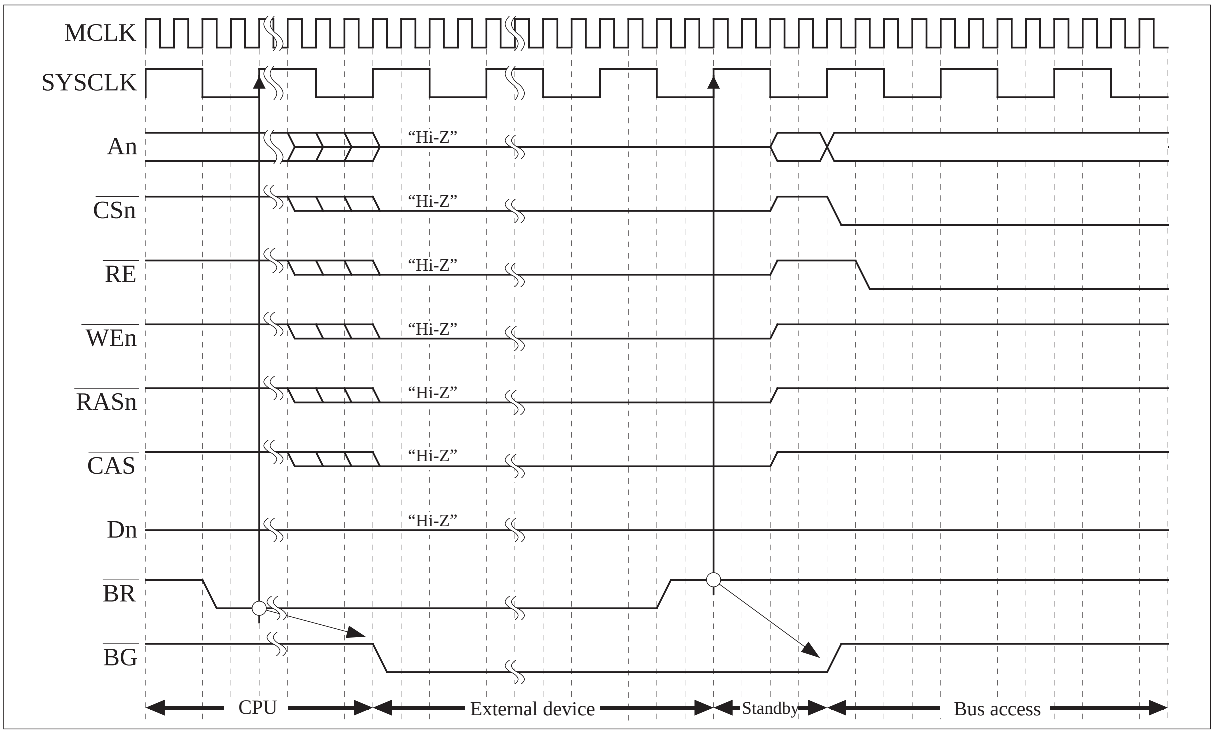


Fig. 8-15-1 Bus Arbitration Timing 1
(Bus Authority Release/Bus Authority Acquisition, nfr = 4)

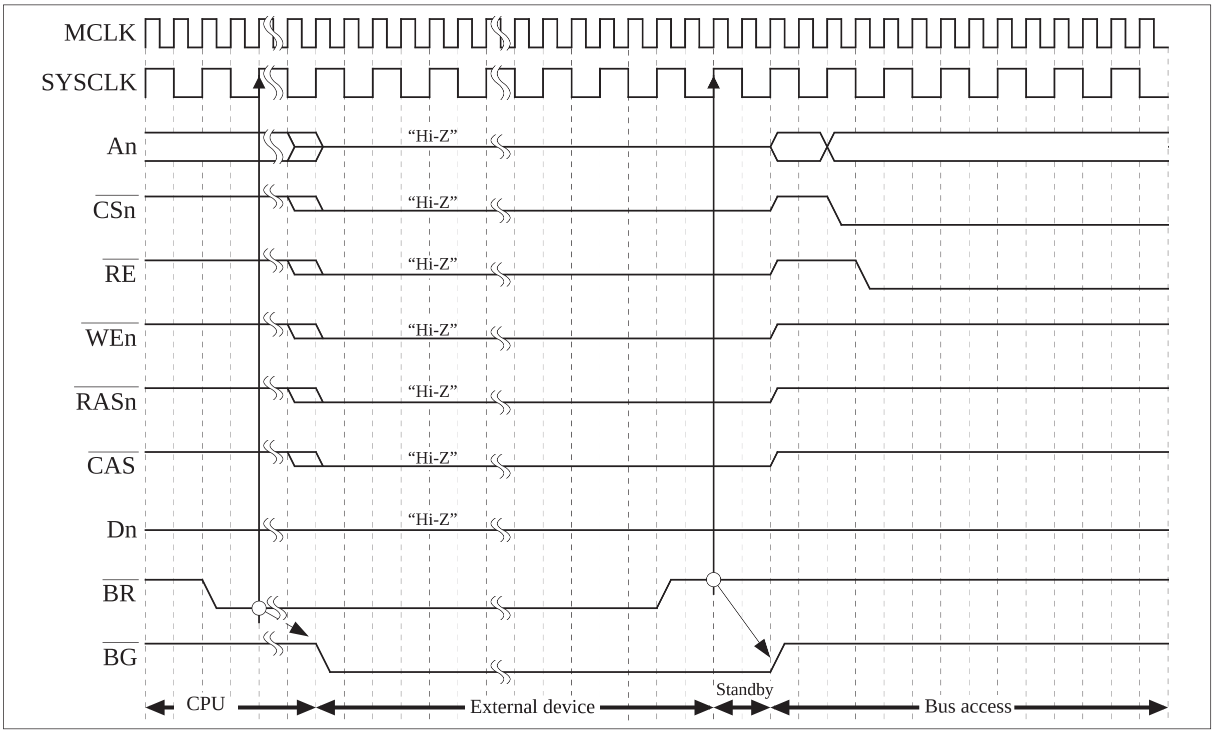


Fig. 8-15-2 Bus Arbitration Timing 2
(Bus Authority Release/Bus Authority Acquisition, nfr = 2)

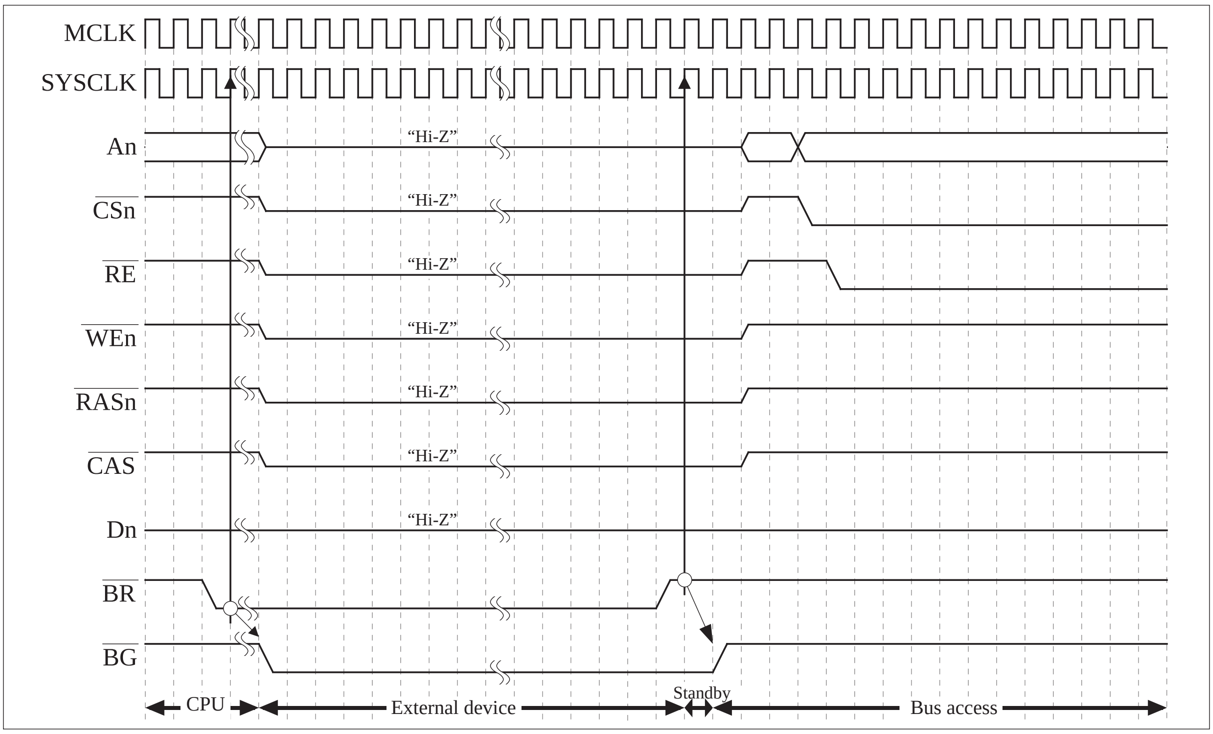


Fig. 8-15-3 Bus Arbitration Timing 3
(Bus Authority Release/Bus Authority Acquisition, nfr = 1)

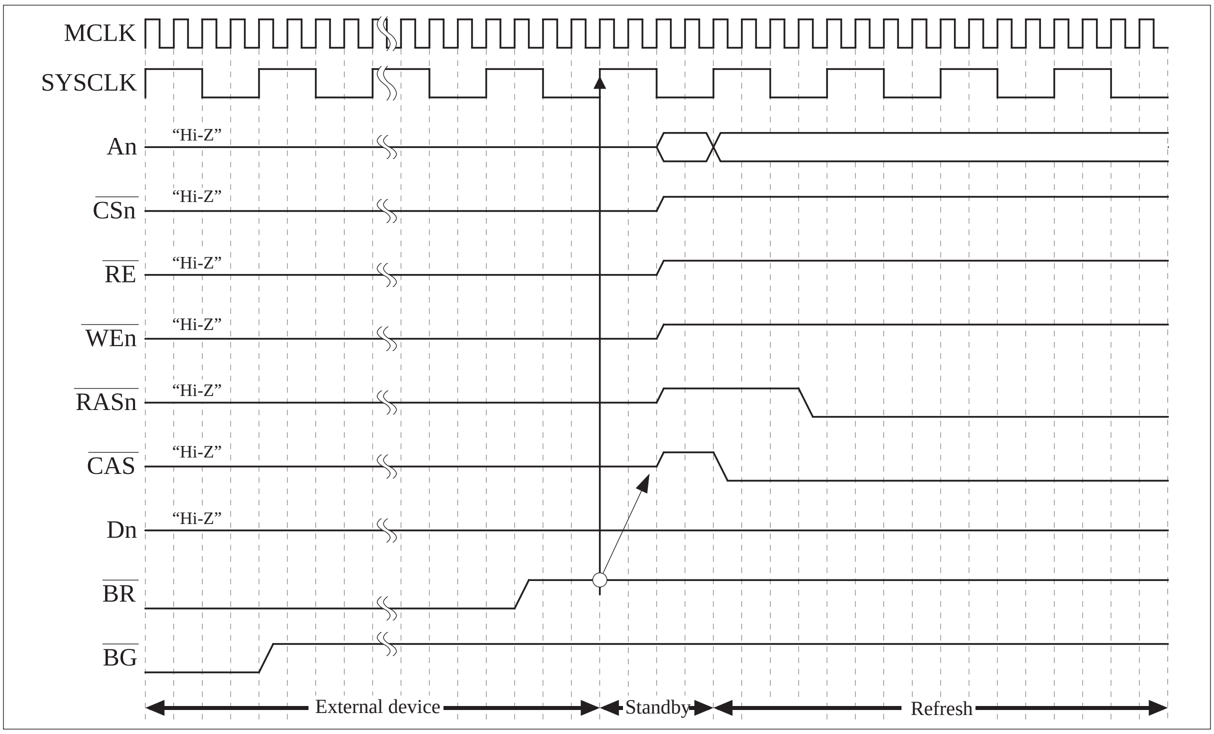


Fig. 8-15-4 Bus Arbitration Timing 4
(Refresh Request Generated While Bus Authority Has Been Released)

8.16 Cautions

These cautions concern the BC. These cautions must be heeded, since failure to do so may result in misoperation.

1. Do not change the contents of the relevant memory control register and the DRAM control register while accessing external memory space, except when software page mode is not in effect.

2. Do not overwrite the refresh counter register while the REFE bit is “1” in the DRAM control register.

3. “0” is output on pins A23* to 16 when pins ADM15 to 0 are operating as data pins in address/data multiplex mode, as shown in the diagram below; as a result, in order to use pins A23* to 16 while pins ADM15 to 0 are operating as data pins, it is necessary to latch the output on pins A23* to 16 with the address strobe AS.

*: A23 also serves as $\overline{\text{CS3}}$.

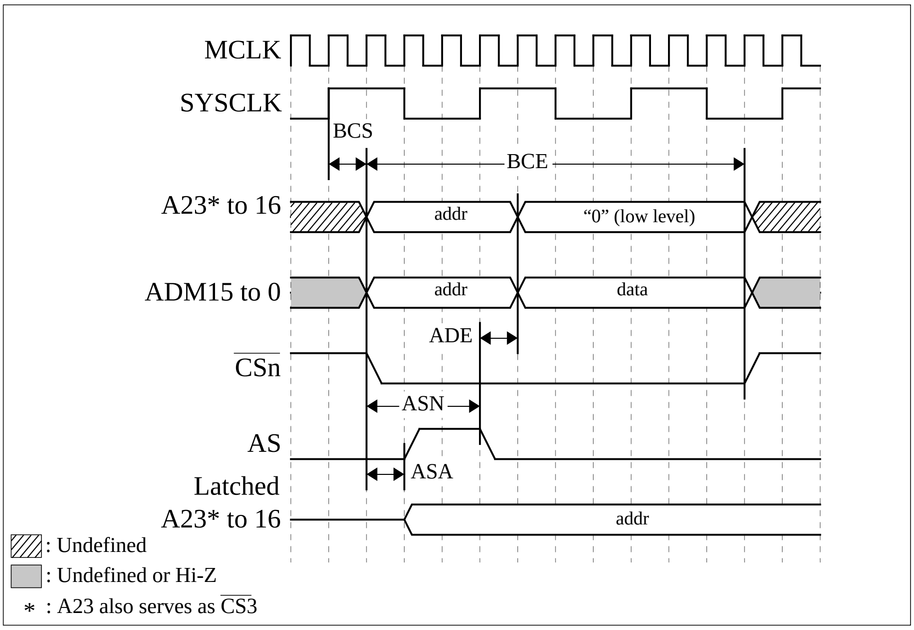


Fig. 8-16-1 Example of Address Pin Usage in Address/Data Multiplex Mode

4. When entering the stop mode, the output mode of pins ADM15 to 0 is undefined. Therefore, when pins ADM15 to 0 are being pulled up according to the I/O port output mode register setting, the pull-up setting for pins ADM15 to 0 should be released before entering the stop mode in order to avoid power consumption in the stop mode due to the pull-up resistance.

   Note: For details on the output mode register settings, refer to Chapter 15, “I/O Ports.”

5. Interrupts are prohibited and the bus is locked (occupied by the CPU) when executing BSET or BCLR, however, if a BSET or BCLR instruction is executed during program execution in external memory, a bus authority release due to an external bus request may be interposed between the data read and data write by the BSET or BCLR instruction.
   If the atomic bus cycles (i.e. bus lock) of the BSET or BCLR instruction need to be guaranteed in a system that uses multiple processors, either of the following measures should be taken.

   1. A program in which a BSET or BCLR instruction is executed should be placed in internal memory.

2. Designate the bus authority request pin ($\overline{\text{BR}}$) as a general-purpose input port, and the bus authority release pin ($\overline{\text{BG}}$) as a general-purpose output port, for instance, so that bus requests cannot be accepted during execution of a BSET or BCLR instruction.

9. Interrupt Controller

9.1 Overview

The interrupt controller processes non-maskable interrupts and level interrupts (internal interrupts and external interrupts).

For external pins, the microcontroller has eight external interrupt pins and one non-maskable interrupt pin.

9.2 Features

- Up to four interrupt requests can be accepted by each group.
- Interrupt priority level: Can be set for each interrupt group.
- External pin interrupt conditions: Positive edge, negative edge, "H" level, "L" level
  Recovery from STOP, HALT or SLEEP mode is possible by means of an external pin interrupt

9.3 System Diagram

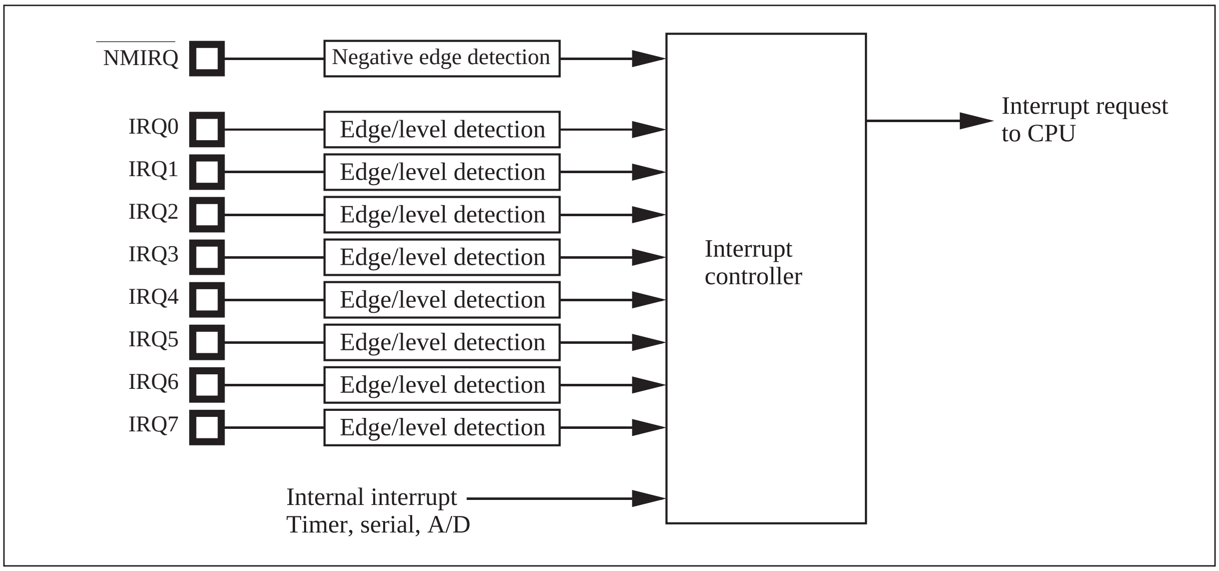
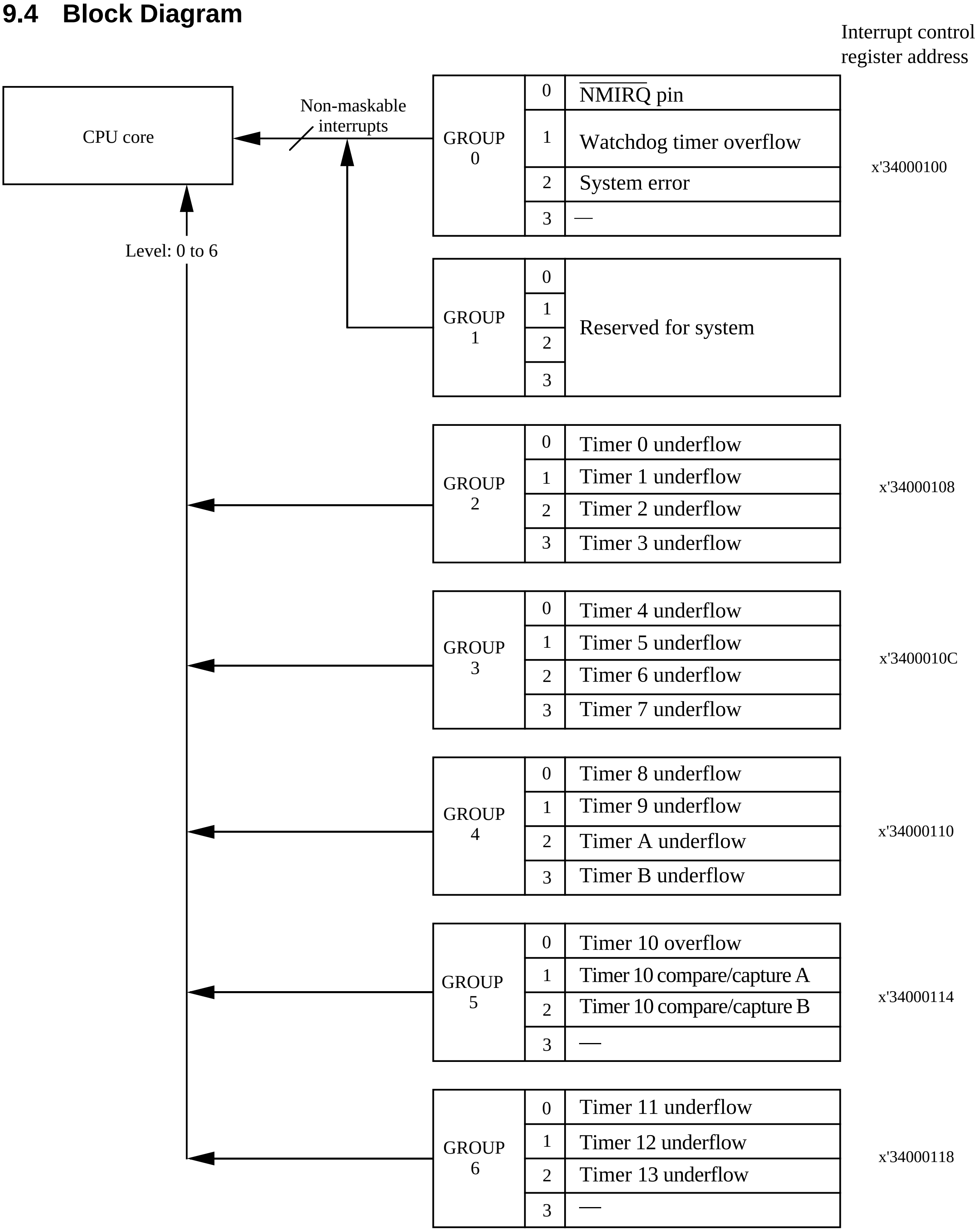


Fig. 9-3-1 System Diagram

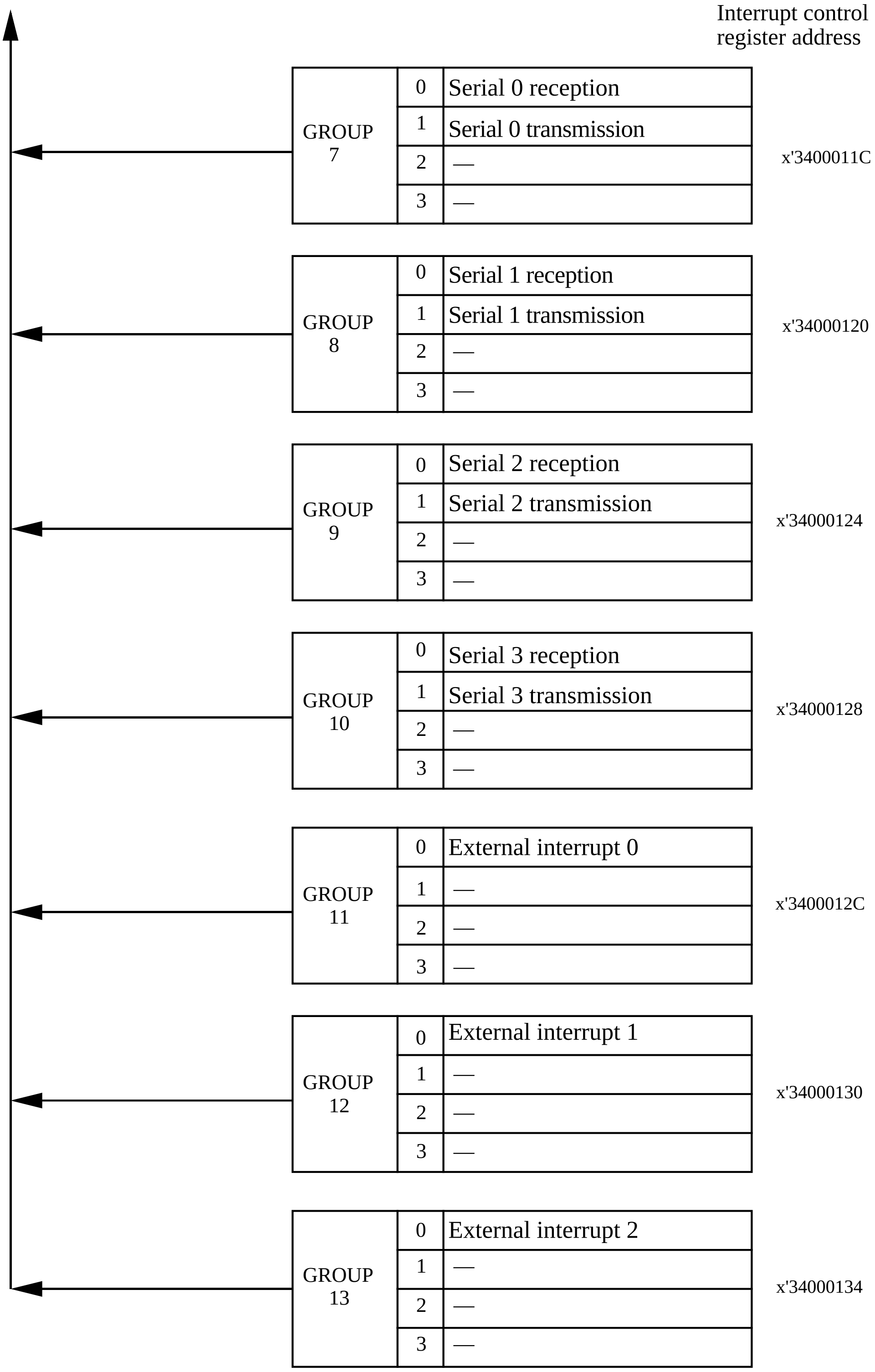
9.4 Block Diagram



			Interrupt control register address
GROUP 0	0	$\overline{\text{NMIRQ}}$ pin	x'34000100
	1	Watchdog timer overflow	
	2	System error	
	3	—	
GROUP 1	0	Reserved for system	
	1		
	2		
	3		
GROUP 2	0	Timer 0 underflow	x'34000108
	1	Timer 1 underflow	
	2	Timer 2 underflow	
	3	Timer 3 underflow	
GROUP 3	0	Timer 4 underflow	x'3400010C
	1	Timer 5 underflow	
	2	Timer 6 underflow	
	3	Timer 7 underflow	
GROUP 4	0	Timer 8 underflow	x'34000110
	1	Timer 9 underflow	
	2	Timer A underflow	
	3	Timer B underflow	
GROUP 5	0	Timer 10 overflow	x'34000114
	1	Timer 10 compare/capture A	
	2	Timer 10 compare/capture B	
	3	—	
GROUP 6	0	Timer 11 underflow	x'34000118
	1	Timer 12 underflow	
	2	Timer 13 underflow	
	3	—	

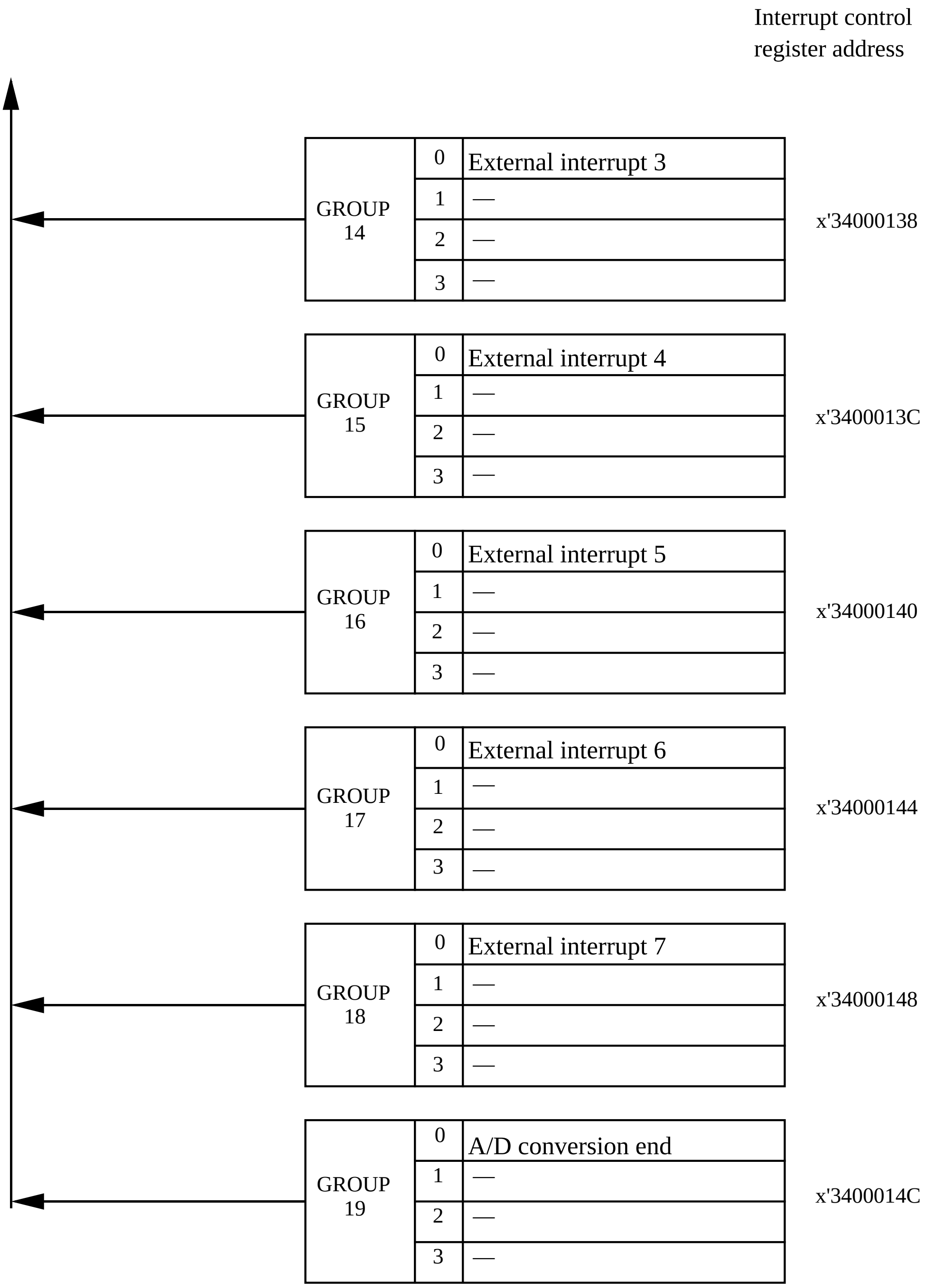
The interrupt level can be set separately for each group. (However, GROUP 0 and GROUP 1 are non-maskable.)

Fig. 9-4-1 Block Diagram 1



			Interrupt control register address
GROUP 7	0	Serial 0 reception	x'3400011C
	1	Serial 0 transmission	
	2	—	
	3	—	
GROUP 8	0	Serial 1 reception	x'34000120
	1	Serial 1 transmission	
	2	—	
	3	—	
GROUP 9	0	Serial 2 reception	x'34000124
	1	Serial 2 transmission	
	2	—	
	3	—	
GROUP 10	0	Serial 3 reception	x'34000128
	1	Serial 3 transmission	
	2	—	
	3	—	
GROUP 11	0	External interrupt 0	x'3400012C
	1	—	
	2	—	
	3	—	
GROUP 12	0	External interrupt 1	x'34000130
	1	—	
	2	—	
	3	—	
GROUP 13	0	External interrupt 2	x'34000134
	1	—	
	2	—	
	3	—	

Fig. 9-4-2   Block Diagram 2



			Interrupt control register address
GROUP 14	0	External interrupt 3	x'34000138
	1	—	
	2	—	
	3	—	
GROUP 15	0	External interrupt 4	x'3400013C
	1	—	
	2	—	
	3	—	
GROUP 16	0	External interrupt 5	x'34000140
	1	—	
	2	—	
	3	—	
GROUP 17	0	External interrupt 6	x'34000144
	1	—	
	2	—	
	3	—	
GROUP 18	0	External interrupt 7	x'34000148
	1	—	
	2	—	
	3	—	
GROUP 19	0	A/D conversion end	x'3400014C
	1	—	
	2	—	
	3	—	

Fig. 9-4-3   Block Diagram 3

9.5 Description of Registers

This interrupt controller includes an interrupt control registers, an interrupt accepted group register, and an external interrupt condition specification register.

Table 9-5-1 lists the interrupt controller registers.

Table 9-5-1 Register List

Address	Name	Symbol	Number of bits	Initial value	Access size
x'34000100	Non-maskable interrupt control register	G0ICR (NMICR)	16	x'0000	8, 16
x'34000108	Group 2 interrupt control register	G2ICR	16	x'0000	8, 16
x'3400010C	Group 3 interrupt control register	G3ICR	16	x'0000	8, 16
x'34000110	Group 4 interrupt control register	G4ICR	16	x'0000	8, 16
x'34000114	Group 5 interrupt control register	G5ICR	16	x'0000	8, 16
x'34000118	Group 6 interrupt control register	G6ICR	16	x'0000	8, 16
x'3400011C	Group 7 interrupt control register	G7ICR	16	x'0000	8, 16
x'34000120	Group 8 interrupt control register	G8ICR	16	x'0000	8, 16
x'34000124	Group 9 interrupt control register	G9ICR	16	x'0000	8, 16
x'34000128	Group 10 interrupt control register	G10ICR	16	x'0000	8, 16
x'3400012C	Group 11 interrupt control register	G11ICR	16	x'0000	8, 16
x'34000130	Group 12 interrupt control register	G12ICR	16	x'0000	8, 16
x'34000134	Group 13 interrupt control register	G13ICR	16	x'0000	8, 16
x'34000138	Group 14 interrupt control register	G14ICR	16	x'0000	8, 16
x'3400013C	Group 15 interrupt control register	G15ICR	16	x'0000	8, 16
x'34000140	Group 16 interrupt control register	G16ICR	16	x'0000	8, 16
x'34000144	Group 17 interrupt control register	G17ICR	16	x'0000	8, 16
x'34000148	Group 18 interrupt control register	G18ICR	16	x'0000	8, 16
x'3400014C	Group 19 interrupt control register	G19ICR	16	x'0000	8, 16
x'34000200	Interrupt accepted group register	IAGR	16	x'0000	8, 16
x'34000280	External interrupt condition specification register	EXTMD	16	x'0000	8, 16

Non-maskable interrupt control register

Register symbol: G0ICR (NMICR)
Address: x'34000100
Purpose: This register determines whether a non-maskable interrupt has been generated.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	-	-	-	-	-	-	-	-	-	-	-	-	-	SYSEF	WDIF	NMIF
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit No.	Bit name	Description
0	NMIF	External non-maskable interrupt request flag 0: No interrupt request 1: Interrupt detected
1	WDIF	Watchdog timer overflow interrupt request flag 0: No interrupt request 1: Interrupt detected
2	SYSEF	System error interrupt request flag 0: No interrupt request 1: Interrupt detected

The method of clearing flag differs according to the interrupt request flags.

1. External non-maskable interrupt request flag (NMIF) and Watchdog timer overflow interrupt request flag (WDIF)

After a non-maskable interrupt is accepted, these flags can be cleared by writing to the non-maskable interrupt control register (NMICR).
When a flag is set to “1”, write a “1” to the flag to clear it.

The relationship between the flag status, the data written to the flag, and the new flag status after the data is written is shown in the table below.

Flag status	Write data	Flag status after write	Flag change
0	0	0	No change.
0	1	0	No change.
1	0	1	No change.
1	1	0	Flag is cleared.

Note: A non-maskable interrupt cannot be generated through software.

2. System error interrupt request flag (SYSEF)

This flag cannot be cleared by writing to the non-maskable interrupt control register (NMICR).
This flag can be cleared by generating a reset interrupt by setting the $\overline{\text{RST}}$ pin to "L" level or by the self-reset, which is generated by writing to the reset control register (RSTCTR) of the watchdog timer.

Note*:
Normally, the serial debugger uses non-maskable interrupts. The following points should be observed when performing user application processing using non-maskable interrupts, and when using the serial debugger:

- The non-maskable interrupt processing program should be written so that interrupt factors are checked within the non-maskable interrupt processing program, and control branches to the debugger program for all factors other than those that should be processed.
- The program should be written so that if the stack pointer (SP) is changed before branching to the debugger program, control should branch after the stack pointer is set to the value when the interrupt was accepted.

*For details, refer to the“ MN1030 Series C Source Code Debugger User's Manual”.

Group n interrupt control register GnICR (n = 2 to 19)

Registers G2ICR to G19ICR control level interrupts for groups 2 to 19, respectively.

Each register confirms the group interrupt level as well as the enabling, request, and detection of interrupts within the respective group.

The explanation on this page applies to registers G2ICR to G19ICR.

The interrupt control registers for group 2 to 19 are described starting on page 9-10.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	-	LV2	LV1	LV0	IE3	IE2	IE1	IE0	IR3	IR2	IR1	IR0	ID3	ID2	ID1	ID0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit No. Bit name Description

3 to 0 ID3 to 0 Group n interrupt detection register

- This register stores the logical product of the IEn(n=3 to 0) and IRn(n=3 to 0) bits.
- If an interrupt that is enabled in the IEn bits is generated, the bit corresponding to the interrupt is "1".

7 to 4 IR3 to 0 Group n interrupt request register

- This register stores interrupt requests. Each bit corresponds to an interrupt.
- After receiving the interrupt, these bits are cleared by software in the interrupt processing program.
- When clearing one of these bits, write a "0" to the bit to be cleared and write a "1" to the corresponding IDn(n=3 to 0) bit.

Write data		Result of write
IRn	IDn	IRn
0	0	No change
1	0	No change
0	1	0
1	1	1

Note: n= 0, 1, 2, 3

The value of IDn after the write is the logical product of the value of IEn after the write and the value of IRn after the write.

Bit No.	Bit name	Description
11 to 8	IE3 to 0	Group n interrupt enable register • This register is used to specify whether an interrupt is enabled or not. • When an IEn(n=3 to 0) bit is set to "1", the corresponding interrupt is enabled. • Setting an IEn bit while the corresponding IRn(n=3 to 0) bit is set generates an interrupt.
14 to 12	LV2 to 0	Group n interrupt priority level register • This register sets the interrupt priority levels. • If the interrupt priority level set in the LV2 to 0 bits is smaller than the IM2 to 0 bits in the PSW, interrupts of the corresponding interrupt group are possible. Interrupts in the same interrupt group are all of the interrupt priority level specified by the LV2 to 0 bits. For details on the interrupt factor assigned to each group, refer section 9.4, "Block Diagram." • When simultaneous interrupt requests are generated from more than one interrupt group, the interrupt with the highest interrupt priority level is accepted. In addition, if multiple interrupt priority levels are set in the same level, the interrupt from the group with the smallest group number is accepted.

Perform operations concerning the interrupt priority level bits (LV2 to 0) and the interrupt enable bits (IE3 to 0) in the group n interrupt control register (GnICR) while interrupts are disabled, as shown below.

```
...................
and    0xf7ff, psw      ;   Clears IE in the PSW.
nop                     ;   Insert in order to guarantee that GnICR is accessed after IE has been definitely
nop                     ;   cleared in pipeline fashion.
mov   d0, (GnICR)       ;   Change LV2 to 0 and IE3 to 0.
mov   (GnICR), d0       ;   Insert in order to synchronize with the store buffer.
or     0x0800, psw      ;   Set IE in the PSW.
...................
```

However, while the interrupt handler is running, IE in the PSW is "0," unless IE has been set. Therefore, it is not necessary to explicitly clear the IE bit and disable interrupts.

The nop instructions indicated above can be any instructions, as long as they do not change the IE bit in the PSW, or change LV2 to 0 or IE3 to 0 in GnICR.

In addition, the reason for inserting two nop instructions is to ensure that the minimum number of cycles needed to change the IE bit in the PSW are provided; therefore, any instruction that consumes at least the same number of cycles as two nop instructions may be inserted.

Group 2 interrupt control register

Register symbol: G2ICR
Address: x'34000108
Purpose: This register is used to enable group 2 interrupts, and to confirm interrupt requests and detection.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	-	G2 LV2	G2 LV1	G2 LV0	TM3 IE	TM2 IE	TM1 IE	TM0 IE	TM3 IR	TM2 IR	TM1 IR	TM0 IR	TM3 ID	TM2 ID	TM1 ID	TM0 ID
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit No.	Bit name	Description
0	TM0ID	Timer 0 underflow interrupt detection flag 0: No interrupt detected 1: Interrupt detected
1	TM1ID	Timer 1 underflow interrupt detection flag 0: No interrupt detected 1: Interrupt detected
2	TM2ID	Timer 2 underflow interrupt detection flag 0: No interrupt detected 1: Interrupt detected
3	TM3ID	Timer 3 underflow interrupt detection flag 0: No interrupt detected 1: Interrupt detected
4	TM0IR	Timer 0 underflow interrupt request flag 0: No interrupt request 1: Interrupt request
5	TM1IR	Timer 1 underflow interrupt request flag 0: No interrupt request 1: Interrupt request
6	TM2IR	Timer 2 underflow interrupt request flag 0: No interrupt request 1: Interrupt request
7	TM3IR	Timer 3 underflow interrupt request flag 0: No interrupt request 1: Interrupt request
8	TM0IE	Timer 0 underflow interrupt enable flag 0: Disabled 1: Enabled
9	TM1IE	Timer 1 underflow interrupt enable flag 0: Disabled 1: Enabled
10	TM2IE	Timer 2 underflow interrupt enable flag 0: Disabled 1: Enabled
11	TM3IE	Timer 3 underflow interrupt enable flag 0: Disabled 1: Enabled
12	G2LV0	Group 2 interrupt priority level register (LSB)
13	G2LV1	Group 2 interrupt priority level register
14	G2LV2	Group 2 interrupt priority level register (MSB) Set a level from 6 to 0.
15	—	"0" is returned when this bit is read.

Group 3 interrupt control register

Register symbol: G3ICR
Address: x'3400010C
Purpose: This register is used to enable group 3 interrupts, and to confirm interrupt requests and detection.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	-	G3 LV2	G3 LV1	G3 LV0	TM7 IE	TM6 IE	TM5 IE	TM4 IE	TM7 IR	TM6 IR	TM5 IR	TM4 IR	TM7 ID	TM6 ID	TM5 ID	TM4 ID
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit No.	Bit name	Description
0	TM4ID	Timer 4 underflow interrupt detection flag
		0: No interrupt detected 1: Interrupt detected
1	TM5ID	Timer 5 underflow interrupt detection flag
		0: No interrupt detected 1: Interrupt detected
2	TM6ID	Timer 6 underflow interrupt detection flag
		0: No interrupt detected 1: Interrupt detected
3	TM7ID	Timer 7 underflow interrupt detection flag
		0: No interrupt detected 1: Interrupt detected
4	TM4IR	Timer 4 underflow interrupt request flag
		0: No interrupt request 1: Interrupt request
5	TM5IR	Timer 5 underflow interrupt request flag
		0: No interrupt request 1: Interrupt request
6	TM6IR	Timer 6 underflow interrupt request flag
		0: No interrupt request 1: Interrupt request
7	TM7IR	Timer 7 underflow interrupt request flag
		0: No interrupt request 1: Interrupt request
8	TM4IE	Timer 4 underflow interrupt enable flag
		0: Disabled 1: Enabled
9	TM5IE	Timer 5 underflow interrupt enable flag
		0: Disabled 1: Enabled
10	TM6IE	Timer 6 underflow interrupt enable flag
		0: Disabled 1: Enabled
11	TM7IE	Timer 7 underflow interrupt enable flag
		0: Disabled 1: Enabled
12	G3LV0	Group 3 interrupt priority level register (LSB)
13	G3LV1	Group 3 interrupt priority level register
14	G3LV2	Group 3 interrupt priority level register (MSB)
		Set a level from 6 to 0.
15	—	"0" is returned when this bit is read.

Group 4 interrupt control register

Register symbol: G4ICR

Address: x'34000110

Purpose: This register is used to enable group 4 interrupts, and to confirm interrupt requests and detection.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	-	G4 LV2	G4 LV1	G4 LV0	TMB IE	TMA IE	TM9 IE	TM8 IE	TMB IR	TMA IR	TM9 IR	TM8 IR	TMB ID	TMA ID	TM9 ID	TM8 ID
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit No. Bit name Description

0 TM8ID Timer 8 underflow interrupt detection flag
0: No interrupt detected 1: Interrupt detected

1 TM9ID Timer 9 underflow interrupt detection flag
0: No interrupt detected 1: Interrupt detected

2 TMAID Timer A underflow interrupt detection flag
0: No interrupt detected 1: Interrupt detected

3 TMBID Timer B underflow interrupt detection flag
0: No interrupt detected 1: Interrupt detected

4 TM8IR Timer 8 underflow interrupt request flag
0: No interrupt request 1: Interrupt request

5 TM9IR Timer 9 underflow interrupt request flag
0: No interrupt request 1: Interrupt request

6 TMAIR Timer A underflow interrupt request flag
0: No interrupt request 1: Interrupt request

7 TMBIR Timer B underflow interrupt request flag
0: No interrupt request 1: Interrupt request

8 TM8IE Timer 8 underflow interrupt enable flag
0: Disabled 1: Enabled

9 TM9IE Timer 9 underflow interrupt enable flag
0: Disabled 1: Enabled

10 TMAIE Timer A underflow interrupt enable flag
0: Disabled 1: Enabled

11 TMBIE Timer B underflow interrupt enable flag
0: Disabled 1: Enabled

12 G4LV0 Group 4 interrupt priority level register (LSB)

13 G4LV1 Group 4 interrupt priority level register

14 G4LV2 Group 4 interrupt priority level register (MSB)
Set a level from 6 to 0.

15 — "0" is returned when this bit is read.

Group 5 interrupt control register

Register symbol: G5ICR

Address: x'34000114

Purpose: This register is used to enable group 5 interrupts, and to confirm interrupt requests and detection.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	-	G5 LV2	G5 LV1	G5 LV0	-	T10B IE	T10A IE	T10U IE	-	T10B IR	T10A IR	T10U IR	-	T10B ID	T10A ID	T10U ID
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit No.	Bit name	Description
0	T10UID	Timer 10 overflow interrupt detection flag 0: No interrupt detected 1: Interrupt detected
1	T10AID	Timer 10 compare/capture A interrupt detection flag 0: No interrupt detected 1: Interrupt detected
2	T10BID	Timer 10 compare/capture B interrupt detection flag 0: No interrupt detected 1: Interrupt detected
3	—	"0" is returned when this bit is read.
4	T10UIR	Timer 10 overflow interrupt request flag 0: No interrupt request 1: Interrupt request
5	T10AIR	Timer 10 compare/capture A interrupt request flag 0: No interrupt request 1: Interrupt request
6	T10BIR	Timer 10 compare/capture B interrupt request flag 0: No interrupt request 1: Interrupt request
7	—	"0" is returned when this bit is read.
8	T10UIE	Timer 10 overflow interrupt enable flag 0: Disabled 1: Enabled
9	T10AIE	Timer 10 compare/capture A interrupt enable flag 0: Disabled 1: Enabled
10	T10BIE	Timer 10 compare/capture B interrupt enable flag 0: Disabled 1: Enabled
11	—	"0" is returned when this bit is read.
12	G5LV0	Group 5 interrupt priority level register (LSB)
13	G5LV1	Group 5 interrupt priority level register
14	G5LV2	Group 5 interrupt priority level register (MSB) Set a level from 6 to 0.
15	—	"0" is returned when this bit is read.

Group 6 interrupt control register

Register symbol: G6ICR
Address: x'34000118
Purpose: This register is used to enable group 6 interrupts, and to confirm interrupt requests and detection.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	-	G6 LV2	G6 LV1	G6 LV0	-	T13 IE	T12 IE	T11 IE	-	T13 IR	T12 IR	T11 IR	-	T13 ID	T12 ID	T11 ID
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit No.	Bit name	Description
0	T11ID	Timer 11 underflow interrupt detection flag 0: No interrupt detected 1: Interrupt detected
1	T12ID	Timer 12 underflow interrupt detection flag 0: No interrupt detected 1: Interrupt detected
2	T13ID	Timer 13 underflow interrupt detection flag 0: No interrupt detected 1: Interrupt detected
3	—	"0" is returned when this bit is read.
4	T11IR	Timer 11 underflow interrupt request flag 0: No interrupt request 1: Interrupt request
5	T12IR	Timer 12 underflow interrupt request flag 0: No interrupt request 1: Interrupt request
6	T13IR	Timer 13 underflow interrupt request flag 0: No interrupt request 1: Interrupt request
7	—	"0" is returned when this bit is read.
8	T11IE	Timer 11 underflow interrupt enable flag 0: Disabled 1: Enabled
9	T12IE	Timer 12 underflow interrupt enable flag 0: Disabled 1: Enabled
10	T13IE	Timer 13 underflow interrupt enable flag 0: Disabled 1: Enabled
11	—	"0" is returned when this bit is read.
12	G6LV0	Group 6 interrupt priority level register (LSB)
13	G6LV1	Group 6 interrupt priority level register
14	G6LV2	Group 6 interrupt priority level register (MSB) Set a level from 6 to 0.
15	—	"0" is returned when this bit is read.

Group 7 interrupt control register

Register symbol: G7ICR

Address: x'3400011C

Purpose: This register is used to enable group 7 interrupts, and to confirm interrupt requests and detection.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	-	G7 LV2	G7 LV1	G7 LV0	-	-	SC0T IE	SC0R IE	-	-	SC0T IR	SC0R IR	-	-	SC0T ID	SC0R ID
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R/W	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W

Bit No.	Bit name	Description
0	SC0RID	Serial 0 reception interrupt detection flag 0: No interrupt detected 1: Interrupt detected
1	SC0TID	Serial 0 transmission interrupt detection flag 0: No interrupt detected 1: Interrupt detected
3 and 2	—	"0" is returned when these bits are read.
4	SC0RIR	Serial 0 reception interrupt request flag 0: No interrupt request 1: Interrupt request
5	SC0TIR	Serial 0 transmission interrupt request flag 0: No interrupt request 1: Interrupt request
7 and 6	—	"0" is returned when these bits are read.
8	SC0RIE	Serial 0 reception interrupt enable flag 0: Disabled 1: Enabled
9	SC0TIE	Serial 0 transmission interrupt enable flag 0: Disabled 1: Enabled
11 and 10	—	"0" is returned when these bits are read.
12	G7LV0	Group 7 interrupt priority level register (LSB)
13	G7LV1	Group 7 interrupt priority level register
14	G7LV2	Group 7 interrupt priority level register (MSB) Set a level from 6 to 0.
15	—	"0" is returned when this bit is read.

Group 8 interrupt control register

Register symbol: G8ICR
Address: x'34000120
Purpose: This register is used to enable group 8 interrupts, and to confirm interrupt requests and detection.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	-	G8 LV2	G8 LV1	G8 LV0	-	-	SC1T IE	SC1R IE	-	-	SC1T IR	SC1R IR	-	-	SC1T ID	SC1R ID
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R/W	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W

Bit No.	Bit name	Description
0	SC1RID	Serial 1 reception interrupt detection flag 0: No interrupt detected 1: Interrupt detected
1	SC1TID	Serial 1 transmission interrupt detection flag 0: No interrupt detected 1: Interrupt detected
3 and 2	—	"0" is returned when these bits are read.
4	SC1RIR	Serial 1 reception interrupt request flag 0: No interrupt request 1: Interrupt request
5	SC1TIR	Serial 1 transmission interrupt request flag 0: No interrupt request 1: Interrupt request
7 and 6	—	"0" is returned when these bits are read.
8	SC1RIE	Serial 1 reception interrupt enable flag 0: Disabled 1: Enabled
9	SC1TIE	Serial 1 transmission interrupt enable flag 0: Disabled 1: Enabled
11 and 10	—	"0" is returned when these bits are read.
12	G8LV0	Group 8 interrupt priority level register (LSB)
13	G8LV1	Group 8 interrupt priority level register
14	G8LV2	Group 8 interrupt priority level register (MSB) Set a level from 6 to 0.
15	—	"0" is returned when this bit is read.

Group 9 interrupt control register

Register symbol: G9ICR
Address: x'34000124
Purpose: This register is used to enable group 9 interrupts, and to confirm interrupt requests and detection.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	-	G9 LV2	G9 LV1	G9 LV0	-	-	SC2T IE	SC2R IE	-	-	SC2T IR	SC2R IR	-	-	SC2T ID	SC2R ID
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R/W	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W

Bit No.	Bit name	Description
0	SC2RID	Serial 2 reception interrupt detection flag 0: No interrupt detected 1: Interrupt detected
1	SC2TID	Serial 2 transmission interrupt detection flag 0: No interrupt detected 1: Interrupt detected
3 and 2	—	"0" is returned when these bits are read.
4	SC2RIR	Serial 2 reception interrupt request flag 0: No interrupt request 1: Interrupt request
5	SC2TIR	Serial 2 transmission interrupt request flag 0: No interrupt request 1: Interrupt request
7 and 6	—	"0" is returned when these bits are read.
8	SC2RIE	Serial 2 reception interrupt enable flag 0: Disabled 1: Enabled
9	SC2TIE	Serial 2 transmission interrupt enable flag 0: Disabled 1: Enabled
11 and 10	—	"0" is returned when these bits are read.
12	G9LV0	Group 9 interrupt priority level register (LSB)
13	G9LV1	Group 9 interrupt priority level register
14	G9LV2	Group 9 interrupt priority level register (MSB) Set a level from 6 to 0.
15	—	"0" is returned when this bit is read.

Group 10 interrupt control register

Register symbol: G10ICR
Address: x'34000128
Purpose: This register is used to enable group 10 interrupts, and to confirm interrupt requests and detection.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	-	G10 LV2	G10 LV1	G10 LV0	-	-	SC3T IE	SC3R IE	-	-	SC3T IR	SC3R IR	-	-	SC3T ID	SC3R ID
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R/W	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W

Bit No.	Bit name	Description
0	SC3RID	Serial 3 reception interrupt detection flag 0: No interrupt detected 1: Interrupt detected
1	SC3TID	Serial 3 transmission interrupt detection flag 0: No interrupt detected 1: Interrupt detected
3 and 2	—	"0" is returned when these bits are read.
4	SC3RIR	Serial 3 reception interrupt request flag 0: No interrupt request 1: Interrupt request
5	SC3TIR	Serial 3 transmission interrupt request flag 0: No interrupt request 1: Interrupt request
7 and 6	—	"0" is returned when these bits are read.
8	SC3RIE	Serial 3 reception interrupt enable flag 0: Disabled 1: Enabled
9	SC3TIE	Serial 3 transmission interrupt enable flag 0: Disabled 1: Enabled
11 and 10	—	"0" is returned when these bits are read.
12	G10LV0	Group 10 interrupt priority level register (LSB)
13	G10LV1	Group 10 interrupt priority level register
14	G10LV2	Group 10 interrupt priority level register (MSB) Set a level from 6 to 0.
15	—	"0" is returned when this bit is read.

Group 11 interrupt control register

Register symbol: G11ICR
Address: x'3400012C
Purpose: This register is used to enable group 11 interrupts, and to confirm interrupt requests and detection.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	-	G11 LV2	G11 LV1	G11 LV0	-	-	-	IQ0 IE	-	-	-	IQ0 IR	-	-	-	IQ0 ID
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R/W	R/W	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Bit No.	Bit name	Description
0	IQ0ID	External interrupt 0 interrupt detection flag 0: No interrupt detected 1: Interrupt detected
3 to 1	—	"0" is returned when these bits are read.
4	IQ0IR	External interrupt 0 interrupt request flag 0: No interrupt request 1: Interrupt request
7 to 5	—	"0" is returned when these bits are read.
8	IQ0IE	External interrupt 0 interrupt enable flag 0: Disabled 1: Enabled
11 to 9	—	"0" is returned when these bits are read.
12	G11LV0	Group 11 interrupt priority level register (LSB)
13	G11LV1	Group 11 interrupt priority level register
14	G11LV2	Group 11 interrupt priority level register (MSB) Set a level from 6 to 0.
15	—	"0" is returned when this bit is read.

Group 12 interrupt control register

Register symbol: G12ICR
Address: x'34000130
Purpose: This register is used to enable group 12 interrupts, and to confirm interrupt requests and detection.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	-	G12 LV2	G12 LV1	G12 LV0	-	-	-	IQ1 IE	-	-	-	IQ1 IR	-	-	-	IQ1 ID
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R/W	R/W	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Bit No.	Bit name	Description
0	IQ1ID	External interrupt 1 interrupt detection flag 0: No interrupt detected 1: Interrupt detected
3 to 1	—	"0" is returned when these bits are read.
4	IQ1IR	External interrupt 1 interrupt request flag 0: No interrupt request 1: Interrupt request
7 to 5	—	"0" is returned when these bits are read.
8	IQ1IE	External interrupt 1 interrupt enable flag 0: Disabled 1: Enabled
11 to 9	—	"0" is returned when these bits are read.
12	G12LV0	Group 12 interrupt priority level register (LSB)
13	G12LV1	Group 12 interrupt priority level register
14	G12LV2	Group 12 interrupt priority level register (MSB) Set a level from 6 to 0.
15	—	"0" is returned when this bit is read.

Group 13 interrupt control register

Register symbol: G13ICR
Address: x'34000134
Purpose: This register is used to enable group 13 interrupts, and to confirm interrupt requests and detection.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	-	G13 LV2	G13 LV1	G13 LV0	-	-	-	IQ2 IE	-	-	-	IQ2 IR	-	-	-	IQ2 ID
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R/W	R/W	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Bit No.	Bit name	Description
0	IQ2ID	External interrupt 2 interrupt detection flag 0: No interrupt detected 1: Interrupt detected
3 to 1	—	"0" is returned when these bits are read.
4	IQ2IR	External interrupt 2 interrupt request flag 0: No interrupt request 1: Interrupt request
7 to 5	—	"0" is returned when these bits are read.
8	IQ2IE	External interrupt 2 interrupt enable flag 0: Disabled 1: Enabled
11 to 9	—	"0" is returned when these bits are read.
12	G13LV0	Group 13 interrupt priority level register (LSB)
13	G13LV1	Group 13 interrupt priority level register
14	G13LV2	Group 13 interrupt priority level register (MSB) Set a level from 6 to 0.
15	—	"0" is returned when this bit is read.

Group 14 interrupt control register

Register symbol: G14ICR
Address: x'34000138
Purpose: This register is used to enable group 14 interrupts, and to confirm interrupt requests and detection.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	-	G14 LV2	G14 LV1	G14 LV0	-	-	-	IQ3 IE	-	-	-	IQ3 IR	-	-	-	IQ3 ID
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R/W	R/W	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Bit No.	Bit name	Description
0	IQ3ID	External interrupt 3 interrupt detection flag 0: No interrupt detected 1: Interrupt detected
3 to 1	—	"0" is returned when these bits are read.
4	IQ3IR	External interrupt 3 interrupt request flag 0: No interrupt request 1: Interrupt request
7 to 5	—	"0" is returned when these bits are read.
8	IQ3IE	External interrupt 3 interrupt enable flag 0: Disabled 1: Enabled
11 to 9	—	"0" is returned when these bits are read.
12	G14LV0	Group 14 interrupt priority level register (LSB)
13	G14LV1	Group 14 interrupt priority level register
14	G14LV2	Group 14 interrupt priority level register (MSB) Set a level from 6 to 0.
15	—	"0" is returned when this bit is read.

Group 15 interrupt control register

Register symbol: G15ICR
Address: x'3400013C
Purpose: This register is used to enable group 15 interrupts, and to confirm interrupt requests and detection.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	-	G15 LV2	G15 LV1	G15 LV0	-	-	-	IQ4 IE	-	-	-	IQ4 IR	-	-	-	IQ4 ID
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R/W	R/W	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Bit No.	Bit name	Description
0	IQ4ID	External interrupt 4 interrupt detection flag 0: No interrupt detected 1: Interrupt detected
3 to 1	—	"0" is returned when these bits are read.
4	IQ4IR	External interrupt 4 interrupt request flag 0: No interrupt request 1: Interrupt request
7 to 5	—	"0" is returned when these bits are read.
8	IQ4IE	External interrupt 4 interrupt enable flag 0: Disabled 1: Enabled
11 to 9	—	"0" is returned when these bits are read.
12	G15LV0	Group 15 interrupt priority level register (LSB)
13	G15LV1	Group 15 interrupt priority level register
14	G15LV2	Group 15 interrupt priority level register (MSB) Set a level from 6 to 0.
15	—	"0" is returned when this bit is read.

Group 16 interrupt control register

Register symbol: G16ICR
Address: x'34000140
Purpose: This register is used to enable group 16 interrupts, and to confirm interrupt requests and detection.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	-	G16 LV2	G16 LV1	G16 LV0	-	-	-	IQ5 IE	-	-	-	IQ5 IR	-	-	-	IQ5 ID
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R/W	R/W	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Bit No.	Bit name	Description
0	IQ5ID	External interrupt 5 interrupt detection flag 0: No interrupt detected 1: Interrupt detected
3 to 1	—	"0" is returned when these bits are read.
4	IQ5IR	External interrupt 5 interrupt request flag 0: No interrupt request 1: Interrupt request
7 to 5	—	"0" is returned when these bits are read.
8	IQ5IE	External interrupt 5 interrupt enable flag 0: Disabled 1: Enabled
11 to 9	—	"0" is returned when these bits are read.
12	G16LV0	Group 16 interrupt priority level register (LSB)
13	G16LV1	Group 16 interrupt priority level register
14	G16LV2	Group 16 interrupt priority level register (MSB) Set a level from 6 to 0.
15	—	"0" is returned when this bit is read.

Group 17 interrupt control register

Register symbol: G17ICR

Address: x'34000144

Purpose: This register is used to enable group 17 interrupts, and to confirm interrupt requests and detection.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	-	G17 LV2	G17 LV1	G17 LV0	-	-	-	IQ6 IE	-	-	-	IQ6 IR	-	-	-	IQ6 ID
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R/W	R/W	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Bit No.	Bit name	Description
0	IQ6ID	External interrupt 6 interrupt detection flag 0: No interrupt detected 1: Interrupt detected
3 to 1	—	"0" is returned when these bits are read.
4	IQ6IR	External interrupt 6 interrupt request flag 0: No interrupt request 1: Interrupt request
7 to 5	—	"0" is returned when these bits are read.
8	IQ6IE	External interrupt 6 interrupt enable flag 0: Disabled 1: Enabled
11 to 9	—	"0" is returned when these bits are read.
12	G17LV0	Group 17 interrupt priority level register (LSB)
13	G17LV1	Group 17 interrupt priority level register
14	G17LV2	Group 17 interrupt priority level register (MSB) Set a level from 6 to 0.
15	—	"0" is returned when this bit is read.

Group 18 interrupt control register

Register symbol: G18ICR
Address: x'34000148
Purpose: This register is used to enable group 18 interrupts, and to confirm interrupt requests and detection.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	-	G18 LV2	G18 LV1	G18 LV0	-	-	-	IQ7 IE	-	-	-	IQ7 IR	-	-	-	IQ7 ID
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R/W	R/W	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Bit No.	Bit name	Description
0	IQ7ID	External interrupt 7 interrupt detection flag 0: No interrupt detected 1: Interrupt detected
3 to 1	—	"0" is returned when these bits are read.
4	IQ7IR	External interrupt 7 interrupt request flag 0: No interrupt request 1: Interrupt request
7 to 5	—	"0" is returned when these bits are read.
8	IQ7IE	External interrupt 7 interrupt enable flag 0: Disabled 1: Enabled
11 to 9	—	"0" is returned when these bits are read.
12	G18LV0	Group 18 interrupt priority level register (LSB)
13	G18LV1	Group 18 interrupt priority level register
14	G18LV2	Group 18 interrupt priority level register (MSB) Set a level from 6 to 0.
15	—	"0" is returned when this bit is read.

Group 19 interrupt control register

Register symbol: G19ICR
Address: x'3400014C
Purpose: This register is used to enable group 19 interrupts, and to confirm interrupt requests and detection.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	-	G19 LV2	G19 LV1	G19 LV0	-	-	-	AD IE	-	-	-	AD IR	-	-	-	AD ID
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R/W	R/W	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Bit No.	Bit name	Description
0	ADID	A/D conversion end interrupt detection flag 0: No interrupt detected 1: Interrupt detected
3 to 1	—	"0" is returned when these bits are read.
4	ADIR	A/D conversion end interrupt request flag 0: No interrupt request 1: Interrupt request
7 to 5	—	"0" is returned when these bits are read.
8	ADIE	A/D conversion end interrupt enable flag 0: Disabled 1: Enabled
11 to 9	—	"0" is returned when these bits are read.
12	G19LV0	Group 19 interrupt priority level register (LSB)
13	G19LV1	Group 19 interrupt priority level register
14	G19LV2	Group 19 interrupt priority level register (MSB) Set a level from 6 to 0.
15	—	"0" is returned when this bit is read.

Interrupt accepted group register

Register symbol: IAGR
Address: x'34000200
Purpose: This register is used to read the group number that generated the interrupt request.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	-	-	-	-	-	-	-	-	-	GN4	GN3	GN2	GN1	GN0	-	-
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit No.	Bit name	Description
6 to 2	GN4 to 0	Group number register The group number that was accepted is stored in GN 4 to 0.

During a register read, this register returns the smallest group number of the groups that are generating an interrupt of the interrupt levels indicated by IM2 to 0 the PSW.

Because when an interrupt generated, the interrupt levels accepted by the CPU are set in IM2 to 0, this register returns a group number that is generating an interrupt level accepted by the CPU. However, if IM2 to 0 are changed, if the interrupt control register is manipulated, or if a new interrupt factor is generated, the value that this register returns may change even while interrupt processing is in progress.

The interrupt accepted group register IAGR is a read-only register; it cannot be written.

External interrupt condition specification register

Register symbol: EXTMD
Address: x'34000280
Purpose: This register specifies the external interrupt generation conditions. Set the desired level or edge for each pin.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	IR7 TG1	IR7 TG0	IR6 TG1	IR6 TG0	IR5 TG1	IR5 TG0	IR4 TG1	IR4 TG0	IR3 TG1	IR3 TG0	IR2 TG1	IR2 TG0	IR1 TG1	IR1 TG0	IR0 TG1	IR0 TG0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit No.	Bit name	Description
0	IR0TG0	IRQ0 pin trigger condition setting (LSB)
1	IR0TG1	IRQ0 pin trigger condition setting (MSB)
		00: Positive edge 01: Negative edge
		10: "H" level 11: "L" level
2	IR1TG0	IRQ1 pin trigger condition setting (LSB)
3	IR1TG1	IRQ1 pin trigger condition setting (MSB)
		00: Positive edge 01: Negative edge
		10: "H" level 11: "L" level
4	IR2TG0	IRQ2 pin trigger condition setting (LSB)
5	IR2TG1	IRQ2 pin trigger condition setting (MSB)
		00: Positive edge 01: Negative edge
		10: "H" level 11: "L" level
6	IR3TG0	IRQ3 pin trigger condition setting (LSB)
7	IR3TG1	IRQ3 pin trigger condition setting (MSB)
		00: Positive edge 01: Negative edge
		10: "H" level 11: "L" level
8	IR4TG0	IRQ4 pin trigger condition setting (LSB)
9	IR4TG1	IRQ4 pin trigger condition setting (MSB)
		00: Positive edge 01: Negative edge
		10: "H" level 11: "L" level
10	IR5TG0	IRQ5 pin trigger condition setting (LSB)
11	IR5TG1	IRQ5 pin trigger condition setting (MSB)
		00: Positive edge 01: Negative edge
		10: "H" level 11: "L" level
12	IR6TG0	IRQ6 pin trigger condition setting (LSB)
13	IR6TG1	IRQ6 pin trigger condition setting (MSB)
		00: Positive edge 01: Negative edge
		10: "H" level 11: "L" level
14	IR7TG0	IRQ7 pin trigger condition setting (LSB)
15	IR7TG1	IRQ7 pin trigger condition setting (MSB)
		00: Positive edge 01: Negative edge
		10: "H" level 11: "L" level

Note : Change the conditions under which the external interrupt is triggered while the IE bit of the interrupt control register GnICR (n=11 to 18) for each of the groups 11 to 18 is not set to "1" (interrupt enable).

9.6 Description of Operation

The following interrupt processing is performed.

- Non-maskable interrupts: $\overline{\text{NMIRQ}}$ pin interrupt
  Watchdog timer overflow interrupt
  System error interrupt

- Level interrupts

  - Internal interrupts: Peripheral interrupts from timer, serial, A/D

  - External interrupts: External pin interrupts x 8

In the event of a level interrupt, an interrupt group determination is made, and an interrupt request is sent to the CPU.

Once the interrupt signal is received, it is determined to be either a non-maskable interrupt or a level interrupt.
If it is a level interrupt, the interrupt group is determined by deciding to which group the interrupt factor belongs.
Once the interrupt group is determined, the interrupt request is sent by manipulating the interrupt control register (GnICR) for that group in order to notify the CPU of the interrupt group level. The interrupt group number is also set in the interrupt acceptance group register (IAGR).
The interrupt level of a group can be determined by reading the interrupt priority level register LV2 to 0 in the interrupt control register (GnICR).

If multiple level interrupt signals are received, the groups to which each belongs is determined and then the interrupt group with the highest priority level is selected. If the group levels are the same, the group with the smallest group number is selected.

The processing described above is not performed in the case of a non-maskable interrupt; instead, the non-maskable interrupt request is simply sent to the CPU.

[Cautions]

1. Maintain external pin interrupt signals for at least 10, 5, or 2.5 SYSCLK cycles when nfr = (MCLK frequency/ SYSCLK frequency) = 1, 2, or 4, respectively. The interrupt cannot be detected if the signal is not maintained for at least that long.
   However, when recovering from HALT mode in response to an external pin interrupt signal, maintain the signal for at least 22, 11, or 5.5 SYSCLK cycles when nfr = 1, 2, or 4, respectively. Furthermore, when recovering from STOP mode in response to an external pin interrupt signal, maintain the signal for at least 10, 5, 2.5 SYSCLK cycles when nfr = 1, 2, or 4, respectively, if there is just one interrupt factor; if there are multiple interrupt factors, continue the external pin interrupt request until the interrupt factors are confirmed by the interrupt processing program.

2 Although it is possible to recover from STOP, HALT, or SLEEP mode in response to an external pin (IRQ7 to 0) interrupt, the trigger conditions for recovery differ for each mode, as indicated in the table below.

Mode	Trigger conditions for recovery
STOP or HALT mode	"H"/"L" level (recovery in response to edge input is not possible)
SLEEP mode	Positive edge/negative edge/"H" level/"L" level

3 When writing a GnICR register in an interrupt program in order to clear IR and ID and then returning from the interrupt program, in order to gain synchronization with the bus controller store buffer be certain to perform an I/O bus access between the execution of the instruction (movbu, etc.) that is used to write the clear data to the GnICR register and the execution of the instruction to return from the interrupt program.

Example: After clearing a GnICR register, read it again.

```
○ mov    0x0f:b,d0     ; (d0 = clear data)
  movbu  d0,(GnICR)    ; Clears the GnICR flags   (GnICR = address of GnICR register to be cleared)
  movhu  (GnICR),d1    ; I/O bus access           (Reads the GnICR register that was cleared)
  rti                  ; Instruction to return from interrupt program
```

If there is no I/O bus access between the instruction that is used to write the clear data to the GnICR register and the instruction to return from the interrupt program, the return from the interrupt program is not guaranteed.

Misoperation will occur when executing the interrupt program again after returning, especially when the RETURN instruction is described after a clear data write.

```
x mov    0x0f:b,d0     ;
  movbu  d0,(GnICR)    ; Clears the GnICR flags
  rti                  ; Instruction to return from interrupt program
```

10. 8-bit Timers

10.1 Overview

This device has 12 reload timers built in.
All are down counters that can be used as interval timers and event counters.
Eight of the timers are also capable of PWM output.

10.2 Features

The features of the 8-bit timers are described below.

- Clock source: An internal clock or external clock can be selected as the clock source. (Timers 0 to B)
  - Internal clock: IOCLK, 1/8 IOCLK, 1/32 IOCLK, timer 0 to 3 underflow
  - External clock: Counts at the rising edge of the pin input.

  Timers 0 and 8, 1 and 9, 2 and A, and 3 and B share multipurpose pins.
- Cascaded connection: Cascaded connection can be used to form a pure 16-, 24-, or 32-bit timer. Timers 0 to 3, 4 to 7, and 8 to B can be cascaded together.
- Interrupts: An interrupt request is generated when a timer underflow occurs. (Timers 0 to B)
- Timer output: Output of underflow cycle divided in half is possible. (Timers 0 to B) Timers 0 and 8, 1 and 9, 2 and A, and 3 and B share multipurpose pins.
- PWM output: PWM output is permitted. (The cycle and duty ratio can be set.) (Timers 4 to B)

  Resolution: 8 bits maximum

  PWM output cycle: 58.5 kHz (Resolution: 8 bits, IOCLK = 15 MHz)
- Serial interface reference clock generation

  Timers 2, 3, 8, and 9 generate reference clocks for serial interfaces 0 to 3.
- A/D conversion start trigger generation

  Timer 2 generates the A/D conversion start trigger.

10.3 Block Diagram

Fig. 10-3-1 shows a block diagram for timers 0 to 3.
Fig. 10-3-2 shows a block diagram for timers 4 to B.
Figures 10-3-3 to 10-3-6 show connection diagrams for the 8-bit timers.

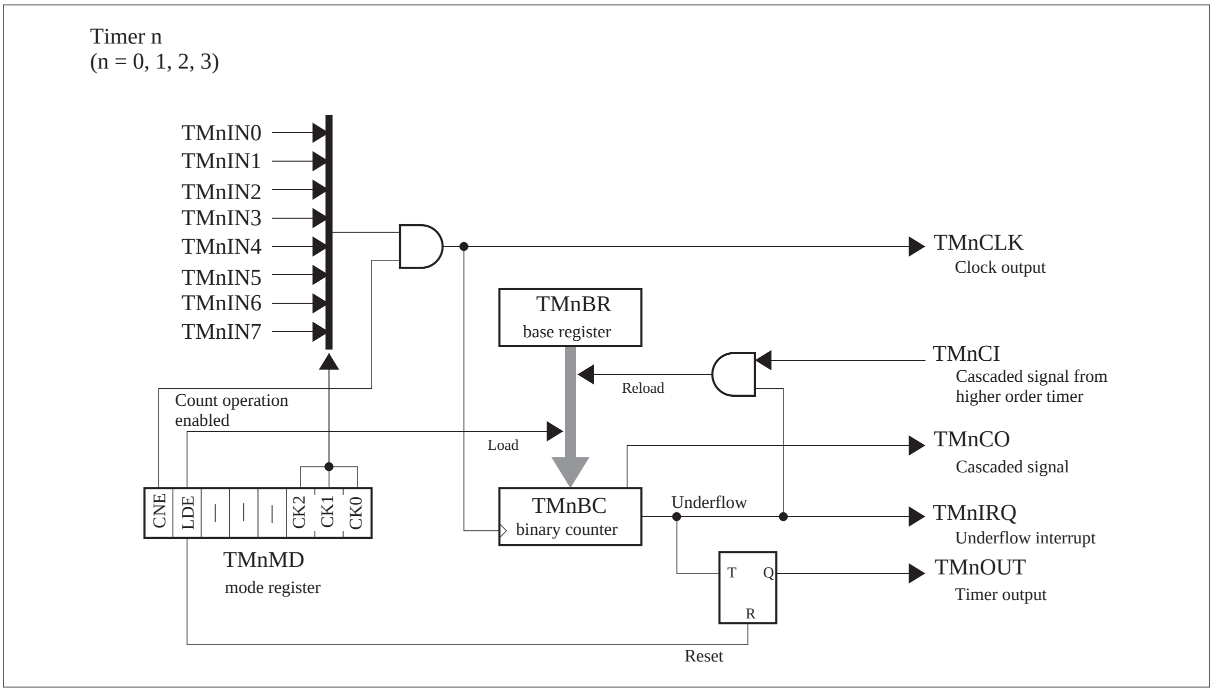


Fig. 10-3-1 8-bit Timer Block Diagram (Timers 0 to 3)

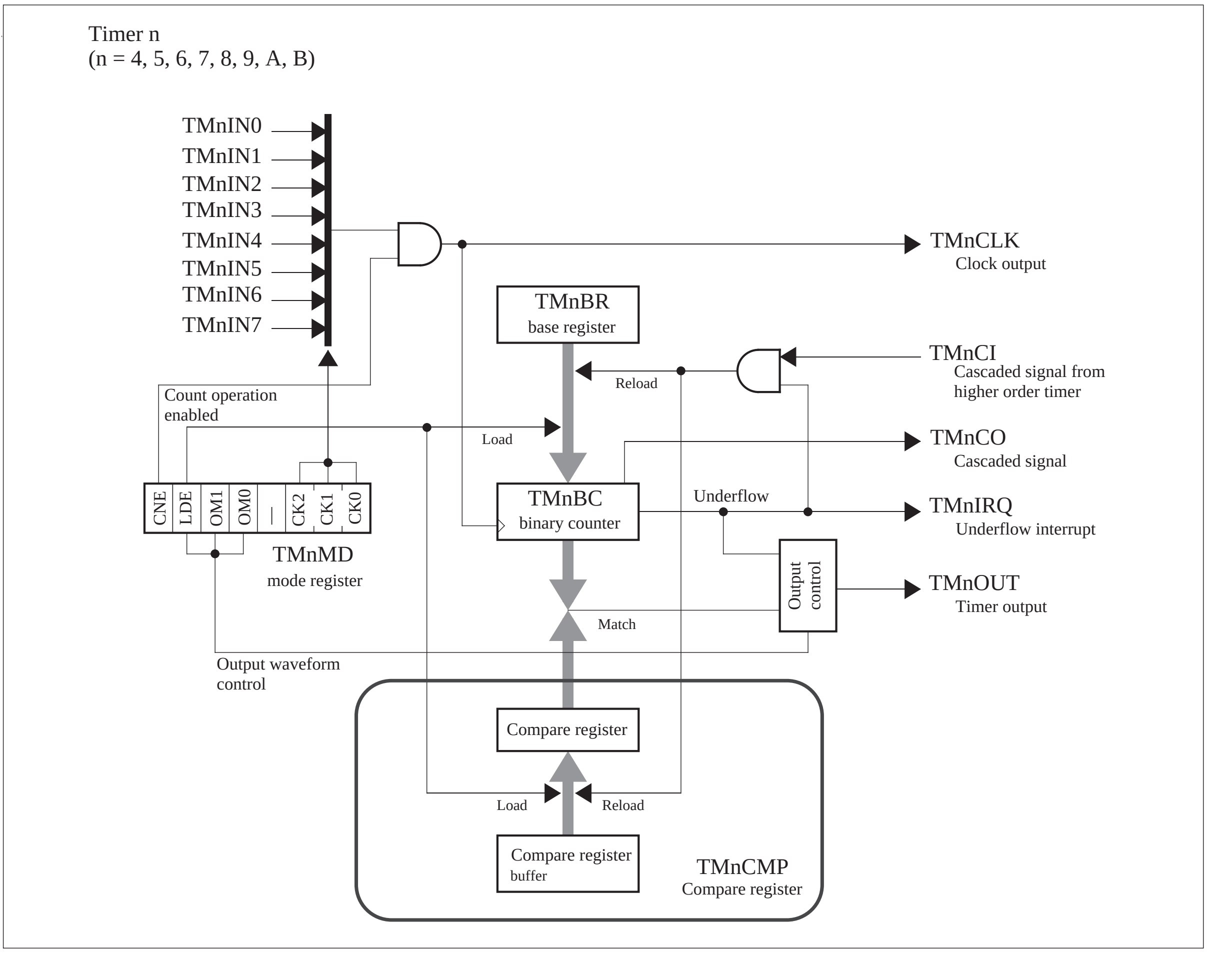


Fig. 10-3-2 8-bit Timer Block Diagram (Timers 4 to B)

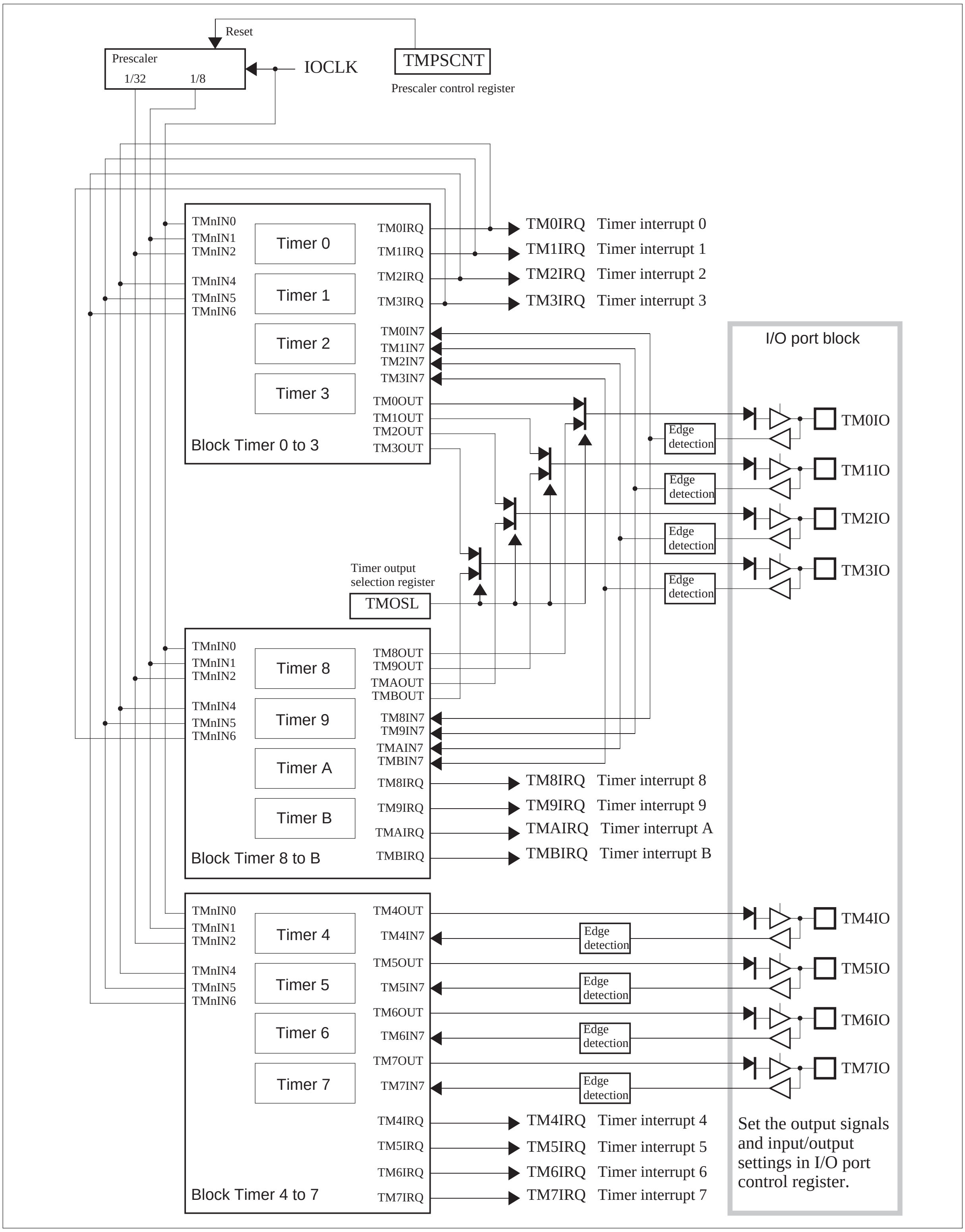


Fig. 10-3-3 8-bit Timer Connection Diagram (Overall)

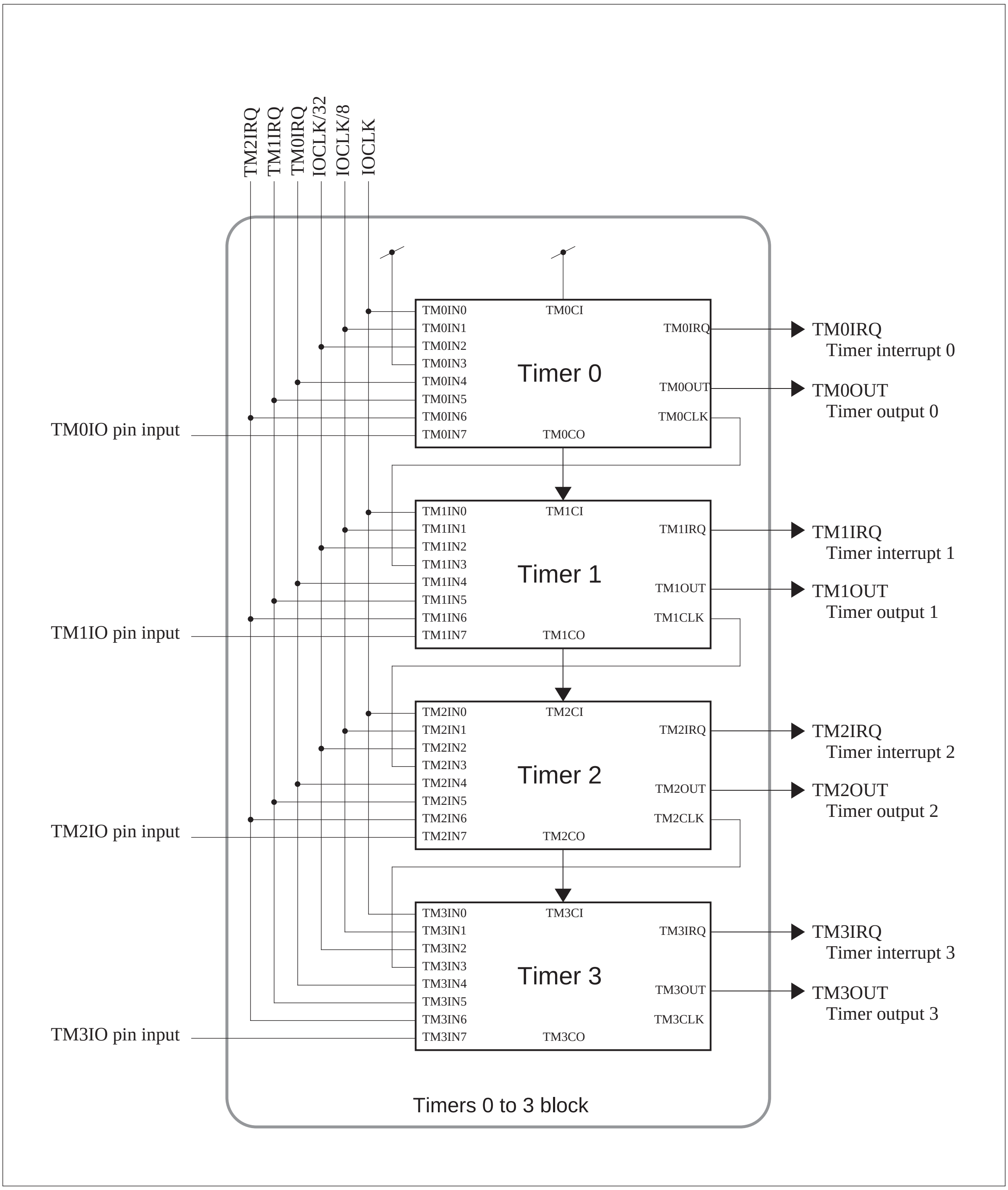


Fig. 10-3-4 8-bit Timer Connection Diagram (Timer 0 to 3 block)

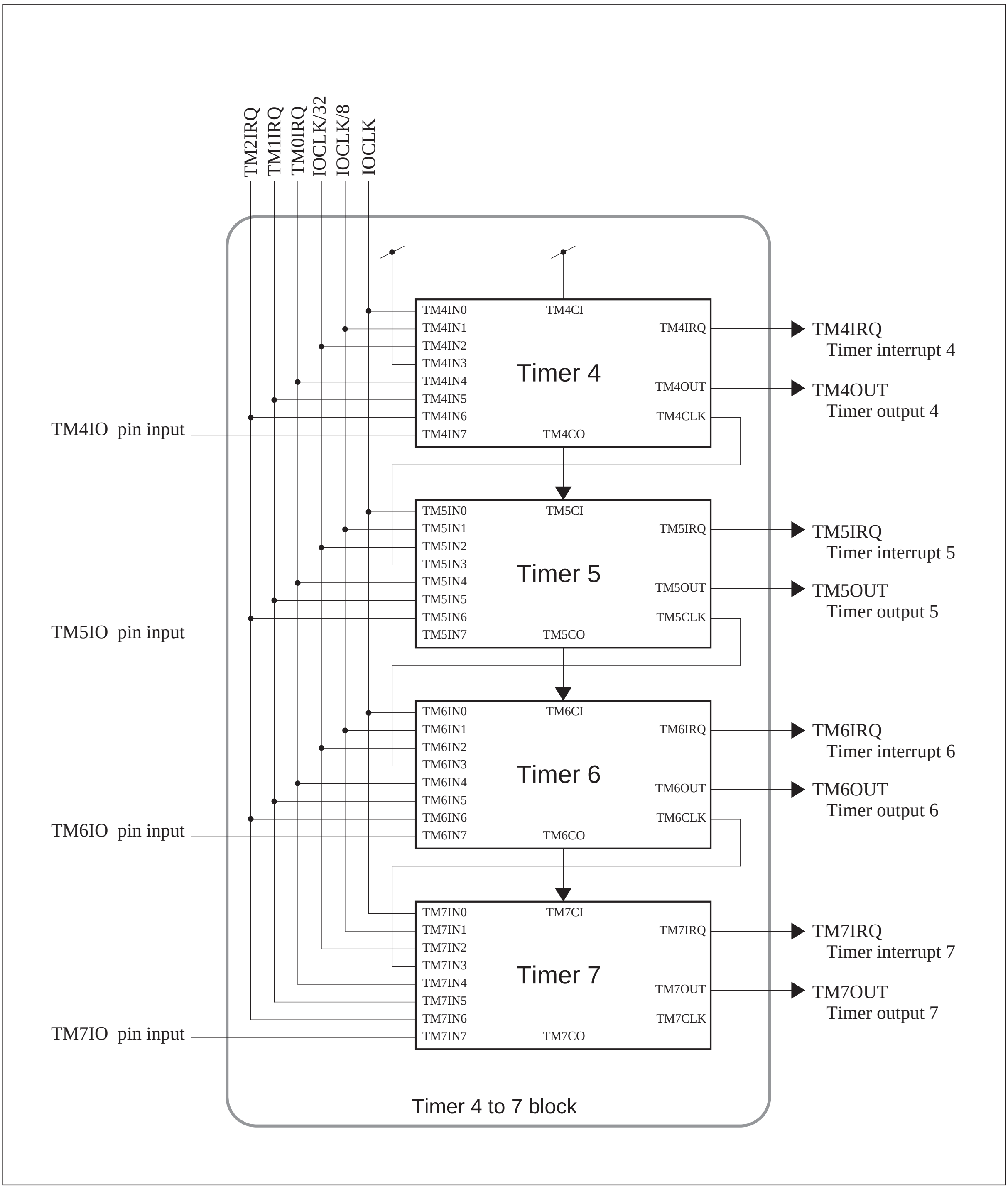


Fig. 10-3-5 8-bit Timer Connection Diagram (Timer 4 to 7 block)

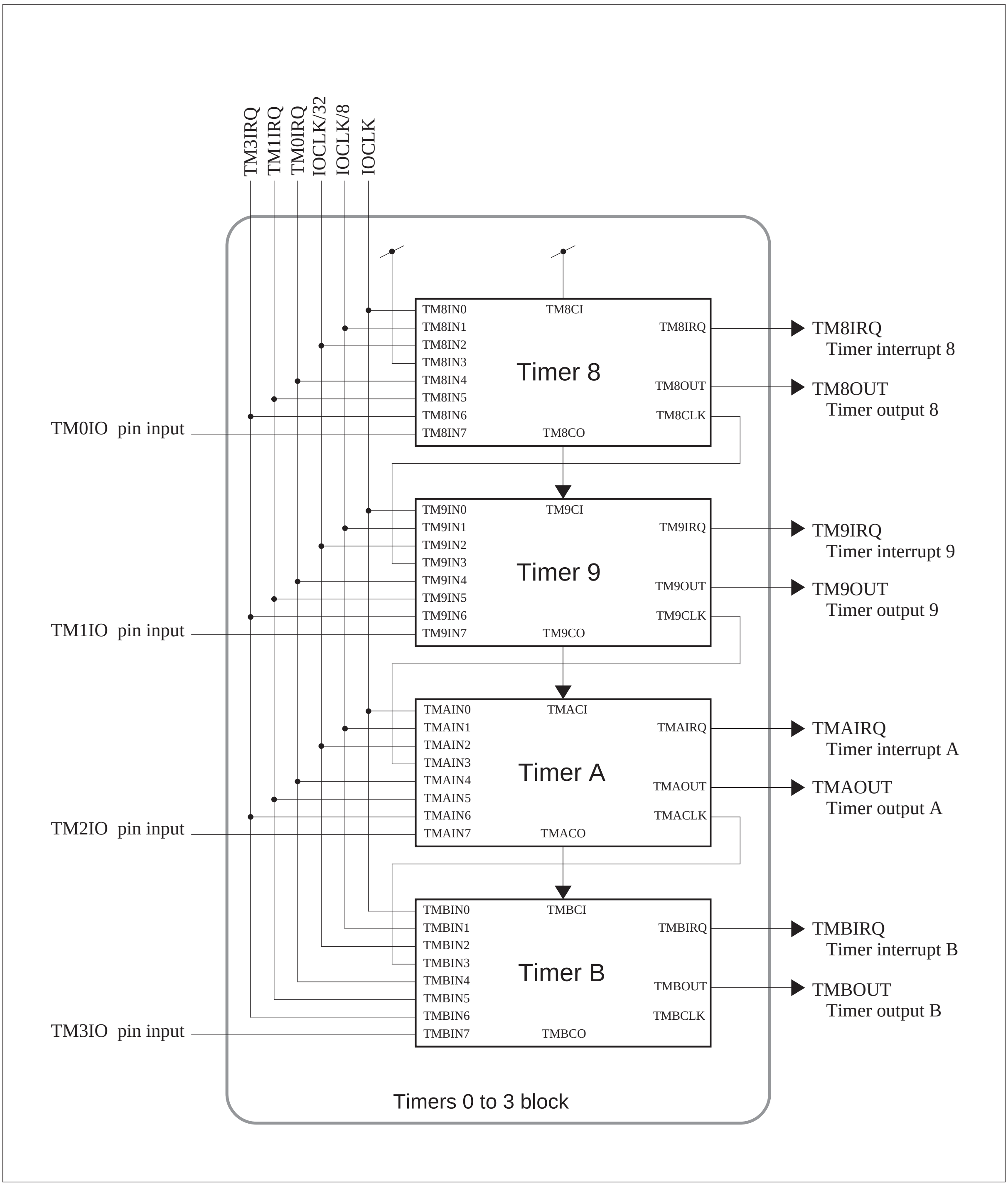


Fig. 10-3-6 8-bit Timer Connection Diagram (Timer 8 to B block)

10.4 Functions

Table 10-4-1 lists the functions of each 8-bit timer.

Table 10-4-1 List of 8-bit Timer Functions

<table>
<tr><th>Timer</th><th colspan="2">0</th><th colspan="2">1</th><th colspan="2">2</th><th colspan="2">3</th><th colspan="2">4</th><th colspan="2">5</th><th colspan="2">6</th><th colspan="2">7</th><th colspan="2">8</th><th colspan="2">9</th><th colspan="2">A</th><th colspan="2">B</th></tr>
<tr><td>Interval timer</td><td colspan="2">✓</td><td colspan="2">✓</td><td colspan="2">✓</td><td colspan="2">✓</td><td colspan="2">✓</td><td colspan="2">✓</td><td colspan="2">✓</td><td colspan="2">✓</td><td colspan="2">✓</td><td colspan="2">✓</td><td colspan="2">✓</td><td colspan="2">✓</td></tr>
<tr><td>Event counter</td><td colspan="2">✓</td><td colspan="2">✓</td><td colspan="2">✓</td><td colspan="2">✓</td><td colspan="2">✓</td><td colspan="2">✓</td><td colspan="2">✓</td><td colspan="2">✓</td><td colspan="2">✓</td><td colspan="2">✓</td><td colspan="2">✓</td><td colspan="2">✓</td></tr>
<tr><td>Timer output</td><td colspan="2">✓</td><td colspan="2">✓</td><td colspan="2">✓</td><td colspan="2">✓</td><td colspan="2">✓</td><td colspan="2">✓</td><td colspan="2">✓</td><td colspan="2">✓</td><td colspan="2">✓</td><td colspan="2">✓</td><td colspan="2">✓</td><td colspan="2">✓</td></tr>
<tr><td>PWM output</td><td colspan="2">—</td><td colspan="2">—</td><td colspan="2">—</td><td colspan="2">—</td><td colspan="2">✓</td><td colspan="2">✓</td><td colspan="2">✓</td><td colspan="2">✓</td><td colspan="2">✓</td><td colspan="2">✓</td><td colspan="2">✓</td><td colspan="2">✓</td></tr>
<tr><td>Interrupt</td><td colspan="2">✓</td><td colspan="2">✓</td><td colspan="2">✓</td><td colspan="2">✓</td><td colspan="2">✓</td><td colspan="2">✓</td><td colspan="2">✓</td><td colspan="2">✓</td><td colspan="2">✓</td><td colspan="2">✓</td><td colspan="2">✓</td><td colspan="2">✓</td></tr>
<tr><td>SIF0, 2 clock source</td><td colspan="2">—</td><td colspan="2">—</td><td colspan="2">—</td><td colspan="2">✓</td><td colspan="2">—</td><td colspan="2">—</td><td colspan="2">—</td><td colspan="2">—</td><td colspan="2">—</td><td colspan="2">✓</td><td colspan="2">—</td><td colspan="2">—</td></tr>
<tr><td>SIF1, 3 clock source</td><td colspan="2">—</td><td colspan="2">—</td><td colspan="2">✓</td><td colspan="2">—</td><td colspan="2">—</td><td colspan="2">—</td><td colspan="2">—</td><td colspan="2">—</td><td colspan="2">✓</td><td colspan="2">—</td><td colspan="2">—</td><td colspan="2">—</td></tr>
<tr><td>A/D conversion start trigger</td><td colspan="2">—</td><td colspan="2">—</td><td colspan="2">✓</td><td colspan="2">—</td><td colspan="2">—</td><td colspan="2">—</td><td colspan="2">—</td><td colspan="2">—</td><td colspan="2">—</td><td colspan="2">—</td><td colspan="2">—</td><td colspan="2">—</td></tr>
<tr><td>Cascaded connection</td><td>▨</td><td colspan="2">✓</td><td colspan="2">✓</td><td colspan="2">✓</td><td>▨</td><td>▨</td><td colspan="2">✓</td><td colspan="2">✓</td><td colspan="2">✓</td><td>▨</td><td>▨</td><td colspan="2">✓</td><td colspan="2">✓</td><td colspan="2">✓</td><td>▨</td></tr>
</table>

Note: Because timers 0 and 8, 1 and 9, 2 and A, and 3 and B share multipurpose output pins, only one of either "timer output" or "PWM output" can be selected.

10.5 Description of Registers

Table 10-5-1 lists the 8-bit timer registers.

Table 10-5-1 List of 8-bit Timer Registers (1/2)

Address	Name	Symbol	Number of bits	Initial value	Access size
x'34001000	Timer 0 mode register	TM0MD	8	x'00	8, 16, 32
x'34001001	Timer 1 mode register	TM1MD	8	x'00	8
x'34001002	Timer 2 mode register	TM2MD	8	x'00	8, 16
x'34001003	Timer 3 mode register	TM3MD	8	x'00	8
x'34001004	Timer 4 mode register	TM4MD	8	x'00	8, 16, 32
x'34001005	Timer 5 mode register	TM5MD	8	x'00	8
x'34001006	Timer 6 mode register	TM6MD	8	x'00	8, 16
x'34001007	Timer 7 mode register	TM7MD	8	x'00	8
x'34001008	Timer 8 mode register	TM8MD	8	x'00	8, 16, 32
x'34001009	Timer 9 mode register	TM9MD	8	x'00	8
x'3400100A	Timer A mode register	TMAMD	8	x'00	8, 16
x'3400100B	Timer B mode register	TMBMD	8	x'00	8
x'34001010	Timer 0 base register	TM0BR	8	x'00	8, 16, 32
x'34001011	Timer 1 base register	TM1BR	8	x'00	8
x'34001012	Timer 2 base register	TM2BR	8	x'00	8, 16
x'34001013	Timer 3 base register	TM3BR	8	x'00	8
x'34001014	Timer 4 base register	TM4BR	8	x'00	8, 16, 32
x'34001015	Timer 5 base register	TM5BR	8	x'00	8
x'34001016	Timer 6 base register	TM6BR	8	x'00	8, 16
x'34001017	Timer 7 base register	TM7BR	8	x'00	8
x'34001018	Timer 8 base register	TM8BR	8	x'00	8, 16, 32
x'34001019	Timer 9 base register	TM9BR	8	x'00	8
x'3400101A	Timer A base register	TMABR	8	x'00	8, 16
x'3400101B	Timer B base register	TMBBR	8	x'00	8
x'34001020	Timer 0 binary counter	TM0BC	8	x'00	8, 16, 32
x'34001021	Timer 1 binary counter	TM1BC	8	x'00	8
x'34001022	Timer 2 binary counter	TM2BC	8	x'00	8, 16
x'34001023	Timer 3 binary counter	TM3BC	8	x'00	8
x'34001024	Timer 4 binary counter	TM4BC	8	x'00	8, 16, 32
x'34001025	Timer 5 binary counter	TM5BC	8	x'00	8
x'34001026	Timer 6 binary counter	TM6BC	8	x'00	8, 16
x'34001027	Timer 7 binary counter	TM7BC	8	x'00	8
x'34001028	Timer 8 binary counter	TM8BC	8	x'00	8, 16, 32
x'34001029	Timer 9 binary counter	TM9BC	8	x'00	8
x'3400102A	Timer A binary counter	TMABC	8	x'00	8, 16
x'3400102B	Timer B binary counter	TMBBC	8	x'00	8

Table 10-5-1 List of 8-bit Timer Registers (2/2)

Address	Name	Symbol	Number of bits	Initial value	Access size
x'34001034	Timer 4 compare register	TM4CMP	8	x'00	8, 16, 32
x'34001035	Timer 5 compare register	TM5CMP	8	x'00	8
x'34001036	Timer 6 compare register	TM6CMP	8	x'00	8, 16
x'34001037	Timer 7 compare register	TM7CMP	8	x'00	8
x'34001038	Timer 8 compare register	TM8CMP	8	x'00	8, 16, 32
x'34001039	Timer 9 compare register	TM9CMP	8	x'00	8
x'3400103A	Timer A compare register	TMACMP	8	x'00	8, 16
x'3400103B	Timer B compare register	TMBCMP	8	x'00	8
x'34001070	Timer ouput selection	TMOSL	8	x'00	8, 16
x'34001071	Prescaler control register	TMPSCNT	8	x'00	8

Timer n mode register (n = 0, 1, 2, 3)

Register symbol: TMnMD
Address: x'34001000 (n=0), x'34001001 (n=1),
x'34001002 (n=2), x'34001003 (n=3)
Purpose: This register controls the operation of timer n.

Bit No.	7	6	5	4	3	2	1	0
Bit name	TMn CNE	TMn LDE	-	-	-	TMn CK2	TMn CK1	TMn CK0
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R	R	R	R/W	R/W	R/W

Bit No.	Bit name	Description
0	TMnCK0	Timer n clock source selection flag (LSB)
1	TMnCK1	Timer n clock source selection flag
2	TMnCK2	Timer n clock source selection flag (MSB)

These bits select the timer clock source.
When pin input is selected, the rising edge of the pin input signal is counted.

For details on each timer clock sources, refer to Table 10-5-3, "8-bit Timer Clock Sources."

Bit No.	Bit name	Description
5 to 3	–	"0" is returned when these bits are read.
6	TMnLDE	Timer n initialization flag

Initializes timer n.
0: Normal operation
1: Initialize
Loads the value in TMnBR into TMnBC. Resets timer output n to "L" level.

Bit No.	Bit name	Description
7	TMnCNE	Timer n output enable flag

Enables/disables the timer n count operation.
0: Operation disabled
1: Operation enabled

[Note]
When setting TMnCNE to "1", do so while TMnLDE is set to "0".
When setting TMnLDE to "1", do so while TMnCNE is set to "0".
Operation is not guaranteed if TMnCNE and TMnLDE are both set to "1" at the same time.

Timer n mode register (n = 4, 5, 6, 7, 8, 9, A, B)

Register symbol: TMnMD

Address: x'34001004 (n=4), x'34001005 (n=5), x'34001006 (n=6),
x'34001007 (n=7), x'34001008 (n=8), x'34001009 (n=9),
x'3400100A (n=A), x'3400100B (n=B)

Purpose: This register controls the operation of timer n.

Bit No.	7	6	5	4	3	2	1	0
Bit name	TMn CNE	TMn LDE	TMn OM1	TMn OM0	-	TMn CK2	TMn CK1	TMn CK0
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit No.	Bit name	Description
0	TMnCK0	Timer n clock source selection flag (LSB)
1	TMnCK1	Timer n clock source selection flag
2	TMnCK2	Timer n clock source selection flag (MSB) These bits select the timer clock source. When pin input is selected, the rising edge of the pin input signal is counted. For details on each timer clock sources, refer to Table 10-5-3, "8-bit Timer Clock Sources."
3	—	"0" is returned when this bit is read.
4	TMnOM0	Timer n output mode flag (LSB)
5	TMnOM1	Timer n output mode flag (MSB) These bits select the timer n output waveform. For details on each PWM output waveform, refer to Table 10-5-2, "PWM Output Waves." 00: Underflow 1/2 cycle output ("L" level output during timer n initialization) 01: Underflow 1/2 cycle output ("H" level output during timer n initialization) 10: PWM output ("L" level output during timer n initialization) 11: PWM output ("H" level output during timer n initialization)
6	TMnLDE	Timer n initialization flag Initializes timer n. 0: Normal operation 1: Initialization Loads the value in TMnBR into TMnBC. Resets timer output n. Loads the value in the compare register buffer into the compare register.
7	TMnCNE	Timer n operation enable flag Enables/disables the timer n count operation. 0: Operation disabled 1: Operation enabled

[Note]
When setting TMnCNE to "1", do so while TMnLDE is set to "0".
When setting TMnLDE to "1", do so while TMnCNE is set to "0".
Operation is not guaranteed if TMnCNE and TMnLDE are both set to "1" at the same time.

Table 10-5-2 PWM Output Waves

TMnOM0 setting	Upon initialization	When TMnBC and TMnCMP settings match	In event of an underflow
0	"L" level output	"H" level output	"L" level output
1	"H" level output	"L" level output	"H" level output

Table 10-5-3 8-bit Timer Clock Sources

TMnCK[2:0] Setting	Timer 0	Timer 1	Timer 2	Timer 3
000	IOCLK	IOCLK	IOCLK	IOCLK
001	IOCLK/8	IOCLK/8	IOCLK/8	IOCLK/8
010	IOCLK/32	IOCLK/32	IOCLK/32	IOCLK/32
011	Setting prohibited	Cascaded with timer 0	Cascaded with timer 1	Cascaded with timer 2
100	Setting prohibited	Timer 0 underflow	Timer 0 underflow	Timer 0 underflow
101	Timer 1 underflow	Setting prohibited	Timer 1 underflow	Timer 1 underflow
110	Timer 2 underflow	Timer 2 underflow	Setting prohibited	Timer 2 underflow
111	TM0IO pin input	TM1IO pin input	TM2IO pin input	TM3IO pin input

TMnCK[2:0] Setting	Timer 4	Timer 5	Timer 6	Timer 7
000	IOCLK	IOCLK	IOCLK	IOCLK
001	IOCLK/8	IOCLK/8	IOCLK/8	IOCLK/8
010	IOCLK/32	IOCLK/32	IOCLK/32	IOCLK/32
011	Setting prohibited	Cascaded with timer 4	Cascaded with timer 5	Cascaded with timer 6
100	Timer 0 underflow	Timer 0 underflow	Timer 0 underflow	Timer 0 underflow
101	Timer 1 underflow	Timer 1 underflow	Timer 1 underflow	Timer 1 underflow
110	Timer 2 underflow	Timer 2 underflow	Timer 2 underflow	Timer 2 underflow
111	TM4IO pin input	TM5IO pin input	TM6IO pin input	TM7IO pin input

TMnCK[2:0] Setting	Timer 8	Timer 9	Timer A	Timer B
000	IOCLK	IOCLK	IOCLK	IOCLK
001	IOCLK/8	IOCLK/8	IOCLK/8	IOCLK/8
010	IOCLK/32	IOCLK/32	IOCLK/32	IOCLK/32
011	Setting prohibited	Cascaded with timer 8	Cascaded with timer 9	Cascaded with timer A
100	Timer 0 underflow	Timer 0 underflow	Timer 0 underflow	Timer 0 underflow
101	Timer 1 underflow	Timer 1 underflow	Timer 1 underflow	Timer 1 underflow
110	Timer 3 underflow	Timer 3 underflow	Timer 3 underflow	Timer 3 underflow
111	TM0IO pin input	TM1IO pin input	TM2IO pin input	TM3IO pin input

When using 1/8 IOCLK or 1/32 IOCLK, the prescaler control register (TMPSCNT) must be set.
When TMnIO pin input was selected, the rising edge of the pin input signal is counted.

Timer n base register (n = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B)

Register symbol: TMnBR

Address: x'34001010 (n = 0), x'34001011 (n = 1), x'34001012 (n = 2),
x'34001013 (n = 3), x'34001014 (n = 4), x'34001015 (n = 5),
x'34001016 (n = 6), x'34001017 (n = 7), x'34001018 (n = 8),
x'34001019 (n = 9), x'3400101A (n =A), x'3400101B (n = B)

Purpose: This register sets the initial value of the timer n binary counter and the underflow cycle.

Bit No.	7	6	5	4	3	2	1	0
Bit name	TMn BR7	TMn BR6	TMn BR5	TMn BR4	TMn BR3	TMn BR2	TMn BR1	TMn BR0
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The value that is set in TMnBR is loaded into TMnBC under the following conditions:

(1) When TMnLDE = 1

(2) When an underflow has occurred

TMnBC generates an underflow interrupt every (value set in TMnBR + 1) counts.

When PWM output has been selected for timers 4 to B, the PWM output cycle is set.

Timer n binary counter (n = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B)

Register symbol: TMnBC

Address: x'34001020 (n = 0), x'34001021 (n = 1), x'34001022 (n = 2),
x'34001023 (n = 3), x'34001024 (n = 4), x'34001025 (n = 5),
x'34001026 (n = 6), x'34001027 (n = 7), x'34001028 (n = 8),
x'34001029 (n = 9), x'3400102A (n = A), x'3400102B (n = B)

Purpose: This register is the binary counter for timer n. The counter value can be read from this register.

Bit No.	7	6	5	4	3	2	1	0
Bit name	TMn BC7	TMn BC6	TMn BC5	TMn BC4	TMn BC3	TMn BC2	TMn BC1	TMn BC0
Reset	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R

This is a down counter.

The initial value for this register is the value that is set in TMnBR, and this register generates an underflow and an interrupt request every (value set in TMnBR + 1) counts.

Timer n compare register (n = 4, 5, 6, 7, 8, 9, A, B)

Register symbol: TMnCMP
Address: x'34001034 (n = 4), x'34001035 (n = 5), x'34001036 (n = 6),
x'34001037 (n = 7), x'34001038 (n = 8), x'34001039 (n = 9),
x'3400103A (n = A), x'3400103B (n = B)
Purpose: This is the timer n compare register.

Bit No.	7	6	5	4	3	2	1	0
Bit name	TMn CMP7	TMn CMP6	TMn CMP5	TMn CMP4	TMn CMP3	TMn CMP2	TMn CMP1	TMn CMP0
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

This register sets the PWM output duty ratio.
The duty ratio is (value set in TMnCMP)/(value set in TMnBR + 1).
When data is written to this register, it is written in the compare register buffer.
The set value is loaded from the compare register buffer into the compare register under the following conditions:

(1) When TMnLDE = 1
(2) When an underflow was generated

When this register is read, the value that is actually in the compare register is read.

Timer output selection register

Register symbol: TMOSL
Address: x'34001070
Purpose: This register selects the 8-bit timer output signal.

Bit No.	7	6	5	4	3	2	1	0
Bit name	-	-	-	-	TM OSL3	TM OSL2	TM OSL1	TM OSL0
Reset	0	0	0	0	0	0	0	0
Access	R	R	R	R	R/W	R/W	R/W	R/W

Bit No.	Bit name	Description
0	TMOSL0	Timer output selection flag 0 Selects the output signal for the TM0IO pin. 0: Selects the timer 0 output. 1: Selects the timer 8 output.
1	TMOSL1	Timer output selection flag 1 Selects the output signal for the TM1IO pin. 0: Selects the timer output 1 1: Selects the timer output 9
2	TMOSL2	Timer output selection flag 2 Selects the output signal for the TM2IO pin. 0: Selects the timer output 2 1: Selects the timer output A
3	TMOSL3	Timer output selection flag 3 Selects the output signal for the TM3IO pin. 0: Selects the timer output 3 1: Selects the timer output B
7 to 4	—	"0" is returned when these bits are read.

Prescaler control register

Register symbol: TMPSCNT
Address: x'34001071
Purpose: This register controls the prescaler operation.

Bit No.	7	6	5	4	3	2	1	0
Bit name	TMPS CNE	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
Access	R/W	R	R	R	R	R	R	R

Bit No.	Bit name	Description
6 to 0	–	"0" is returned when these bits are read.
7	TMPSCNE	Prescaler operation enable flag. Enables/disables operation of the 1/8 IOCLK and 1/32 IOCLK prescaler. 0: Operation disabled 1: Operation enabled

The prescaler that is controlled by this register is also used with the 16-bit timers.

10.6 Description of Operation

This section describes the operation of the 8-bit timers.

10.6.1 Interval Timers and Timer Output

When using an 8-bit timer as an interval timer, make the appropriate settings according to the procedure described below.
The timer in question then operates as an interval timer that generates interrupts on the set cycle. (Refer to Figs. 10-6-1 to 10-6-3.)
When using the timers as a 16-, 24- or 32-bit timer by means of a cascaded connection, refer to section 10.6.3, "Cascaded Connection."
The procedure for generating the reference clock for the serial interfaces and for generating the A/D conversion start trigger is the same.

■ Procedure for initiating operation

(1) Set the timer division ratio.
Set the division ratio in TMnBR.
The interrupt request cycle is then:
(value set in TMnBR + 1) x Clock source cycle

(2) Select the clock source.
Select the clock source through TMnCK[2:0] in the TMnMD register. When using either 1/8 IOCLK or 1/32 IOCLK as the clock source, set TMPSCNE to "1" in the TMPSCNT register to enable prescaler operation.

(3) Output mode setting (Applies to timers 4 to B only.)
Set TMnOM[1:0] in the TMnMD register to underflow 1/2 cycle output, and select the polarity after initialization.

(4) Initialize the timer.
Set TMnLDE to "1" in the TMnMD register to initialize timer n. The value set in TMnBR is loaded into TMnBC as the initial value, and the timer output is reset.
After initialization, be certain to set TMnLDE to "0" to return to normal operation mode.

(5) Set the I/O port. (Applies when using timer output.)
Select the output signal in the TMOSL register.
Set the I/O port to the timer output pin.
Select the timer output for the output signal in the I/O port register, and then set it to the output pin.
Note : For details on the I/O port register settings, refer to chapter 15, "I/O Ports."

(6) Enable the timer counting operation.
Once TMnCNE is set to "1" in the TMnMD register, the counting operation starts.

Once the counting operation is enabled, an underflow interrupt request is generated at fixed intervals.
In addition, the pin output is inverted each time that this interrupt is generated, and the value that is set in TMnBR is loaded into TMnBC.
If the value in the TMnBR register is changed while the counting operation is in progress, that value is loaded as the initial value the next time that an underflow is generated, and then the interrupt cycle changes.

■ Procedure for ending operation

(1) Stop the timer counting operation.
Set TMnCNE to "0" in the TMnMD register, stopping the counting operation.

(2) Initialize the timer, if necessary.
If TMnLDE is set to "1" in the TMnBR register, the value that is set in TMnMD is loaded into TMnBC as the initial value, and the timer output is reset. If only the timer is stopped and "1" is not written to TMnLDE, the status of the binary counter and the pin output are maintained as they were before the counting operation was stopped. If TMnCNE is set to "1", the count resumes from the state that was in effect immediately before the counting operation was stopped.

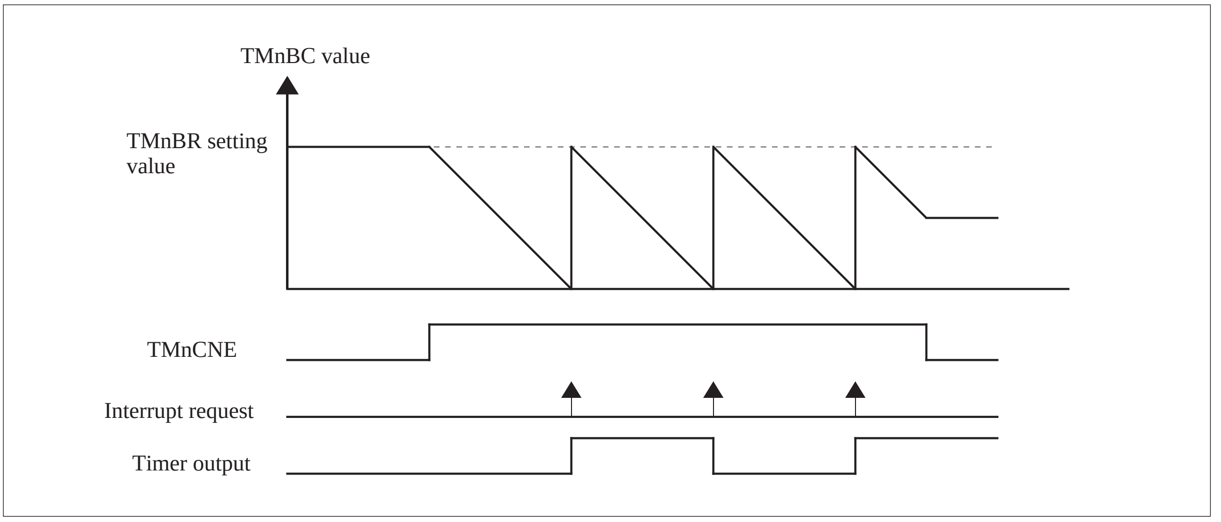


Fig 10-6-1 Interval Timer Operation

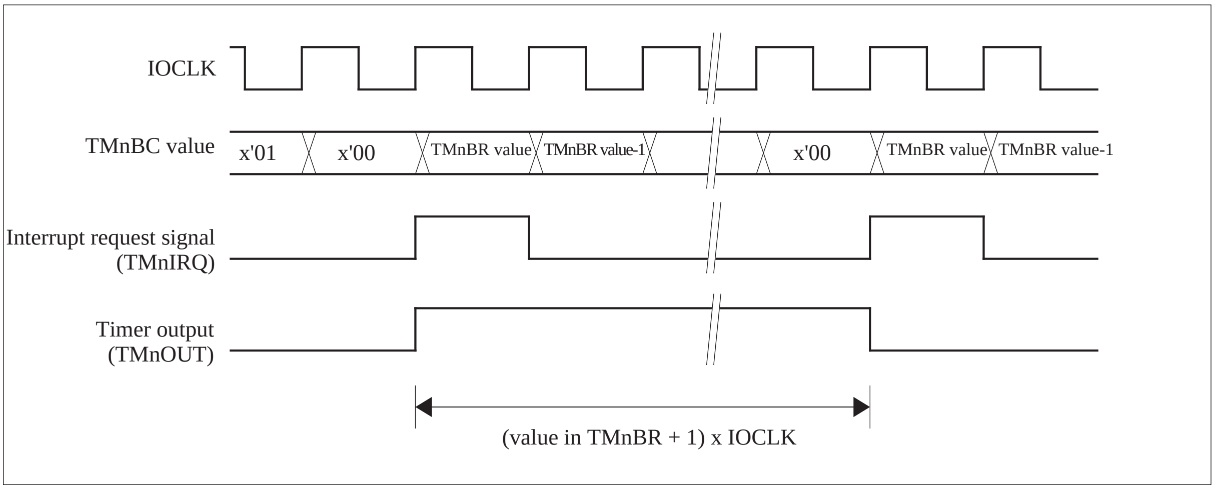


Fig 10-6-2 Interval Timer Operation (When Clock Source = IOCLK)

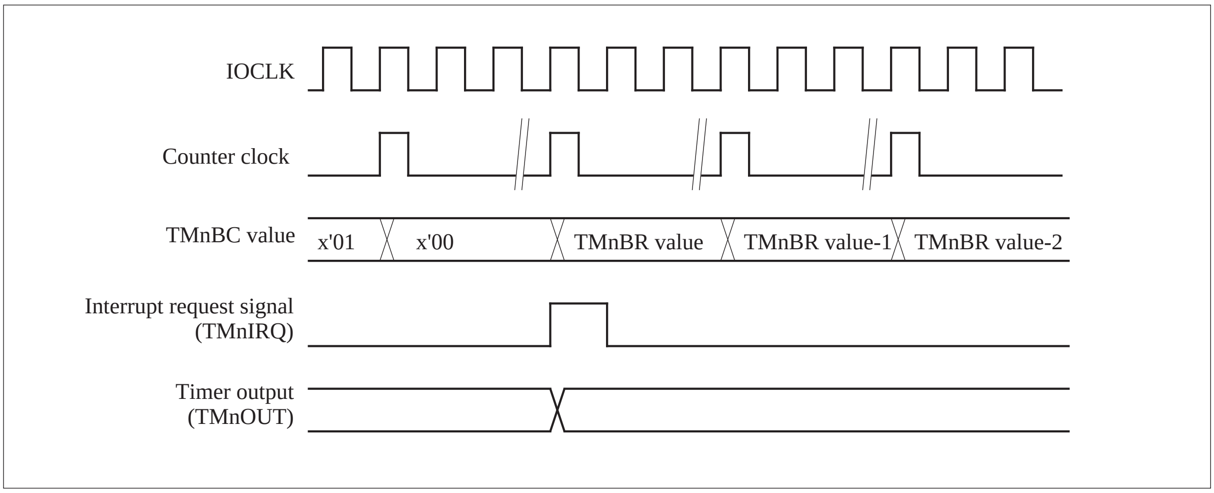


Fig. 10-6-3 Interval Timer Operation (Using Prescaler)

10.6.2 Event Counting

When using an 8-bit timer for event counting, make the settings according to the procedure described below. When using the timers as a 16-, 24- or 32-bit timer by means of a cascaded connection, refer to section 10.6.3, "Cascaded Connection."

■ Procedure for initiating operation

(1) Set the timer division ratio.
Set the division ratio in TMnBR.
An interrupt request is then generated when the rising edge on the pin input is counted (value set in TMnBR + 1) times.

(2) Select the clock source.
Through TMnCK[2:0] in the TMnMD register, set the clock source to the TMnIO pin input.

(3) Initialize the timer.
Set TMnLDE to "1" in the TMnMD register to initialize timer n. The value set in TMnBR is loaded into TMnBC as the initial value.
After initialization, be certain to set TMnLDE to "0" to return to normal operation mode.

(4) Set the I/O port.
Set the I/O port to the general-purpose input pin.
For details on the I/O port register settings, refer to chapter 15, "I/O Ports."

(5) Enable the timer counting operation.
Once TMnCNE is set to "1" in the TMnMD register, the counting operation is enabled.

Once the counting operation is enabled, the rising edge on the pin input is counted, and an interrupt request is generated when there is an underflow in the binary counter. (Refer to Fig. 10-6-4.)
If the value in the TMnBR register is changed while the counting operation is in progress, that value is loaded as the initial value the next time that an underflow is generated.

■ Procedure for ending operation

(1) Stop the timer counting operation.
Set TMnCNE to "0" in the TMnMD register, stopping the counting operation.

(2) Initialize the timer, if necessary.
If TMnLDE is set to "1" in the TMnMD register, the value that is set in TMnBR is loaded into TMnBC as the initial value.
If only the timer is stopped and "1" is not written to TMnLDE, the status of the binary counter is maintained as it was before the counting operation was stopped. If TMnCNE is set to "1", the count resumes from the state that was in effect immediately before the counting operation was stopped.

[Note]
Pin input is sampled according to IOCLK. Input a signal with a pulse width of at least 6, 3, or 1.5 SYSCLK cycles when (MCLK frequency/SYSCLK frequency) = 1, 2, or 4, respectively.
Event counting is not possible when IOCLK is stopped (in HALT or STOP mode).

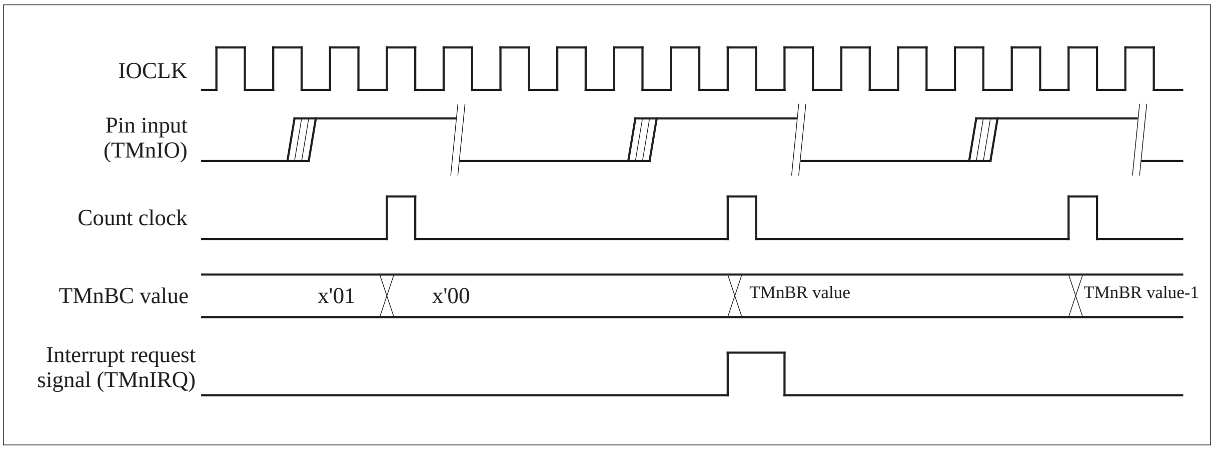


Fig. 10-6-4 Event Counting Operation

10.6.3 Cascaded Connection

The 8-bit timers can be cascaded together in the combinations shown in Fig. 10-6-5.

When using cascaded 8-bit timers as a 16-bit timer:

(Higher)	(Lower)	(Higher)	(Lower)	(Higher)	(Lower)
Timer 3	Timer 2	Timer 2	Timer 1	Timer 1	Timer 0
Timer 7	Timer 6	Timer 6	Timer 5	Timer 5	Timer 4
Timer B	Timer A	Timer A	Timer 9	Timer 9	Timer 8

When using cascaded 8-bit timers as a 24-bit timer:

(Highest)		(Lowest)	(Highest)		(Lowest)
Timer 3	Timer 2	Timer 1	Timer 2	Timer 1	Timer 0
Timer 7	Timer 6	Timer 5	Timer 6	Timer 5	Timer 4
Timer B	Timer A	Timer 9	Timer A	Timer 9	Timer 8

When using cascaded 8-bit timers as a 32-bit timer:

(Highest)			(Lowest)
Timer 3	Timer 2	Timer 1	Timer 0
Timer 7	Timer 6	Timer 5	Timer 4
Timer B	Timer A	Timer 9	Timer 8

Fig. 10-6-5 Cascaded Connection

Make the settings described below when cascading 8-bit timers.

(1) Set the timer division ratio.
Set the timer division ratio in TMnBR.

(Example 1) When using timers 0 and 1 as 16-bit timers and setting the interrupt cycle to x'1234:

In order to set the interrupt cycle to x'1234,
x'1234 - 1 = x'1233
must be set in TMnBR.
Set x'33 in the low-order byte, TM0BR, and x'12 in the high-order byte, TM1BR.

Because TMnBR can be accessed via 16-bit or 32-bit access, values can be set in multiple registers simultaneously. (When cascading timers 1 and 2, 5 and 6, and 9 and A, or when using them as 24-bit timers, it is not possible to simultaneously access only the registers for the cascaded timers.)
When changing the values that are set in TMnBR while the counter is in operation, change TMnBR for the cascaded timers simultaneously.

(2) Select the clock source.
Select any desired clock source for the lowest-order timer.
Set the clock source for the higher timers (all except for the lowest timer) to "cascaded connection."

(Example 1) When using timer 0 and timer 1 as a 16-bit timer
Set the desired clock source for timer 0.
Set the clock source for timer 1 to "cascaded connection."

(Example 2) When using timers 0, 1, 2 and 3 as a 32-bit timer
Set the desired clock source for timer 0.
Set the clock source for timers 1, 2 and 3 to "cascaded connection."

(3) Initialize the timers
Set the TMnLDE flag to "1" for all cascaded timers in order to initialize the timers. (It is not necessary to set the bit simultaneously in all of the registers.)

(4) Enable counting operation

Enable the counting operation by either one of the following two methods:

1) Enable the counting operation for each of the cascaded timers one at a time, in order, starting from the highest timer.
2) Enable the counting operation for all of the cascaded timers simultaneously.

(5) Stop the counting operation

Stop the counting operation by either one of the following two methods:

1) Stop the counting operation for each of the cascaded timers one at a time, in order, starting from the lowest timer.
2) Stop the counting operation for all of the cascaded timers simultaneously.

(6) Timer output and interrupts

Only the timer output and interrupt requests from the highest of the cascaded timers can be used. Operation of the timer output and interrupt requests from the lower cascaded timers is not guaranteed.

⊚Differences between using a timer as a prescaler and when cascaded

The following explanation of these differences uses the cases where the clock source for timer 1 is set to "timer 0 underflow" and to "cascaded with timer 0" as examples.

When "timer 0 underflow" is set, operation is as shown in Fig. 10-6-6. (IOCLK is selected as the clock source for timer 0.)
When TM0BC underflows, the value that is set in TM0BR is loaded into TM0BC, and the value in TM1BC is decremented by one.
When TM1BC underflows, the value that is set in TM1BR is loaded into TM1BC.

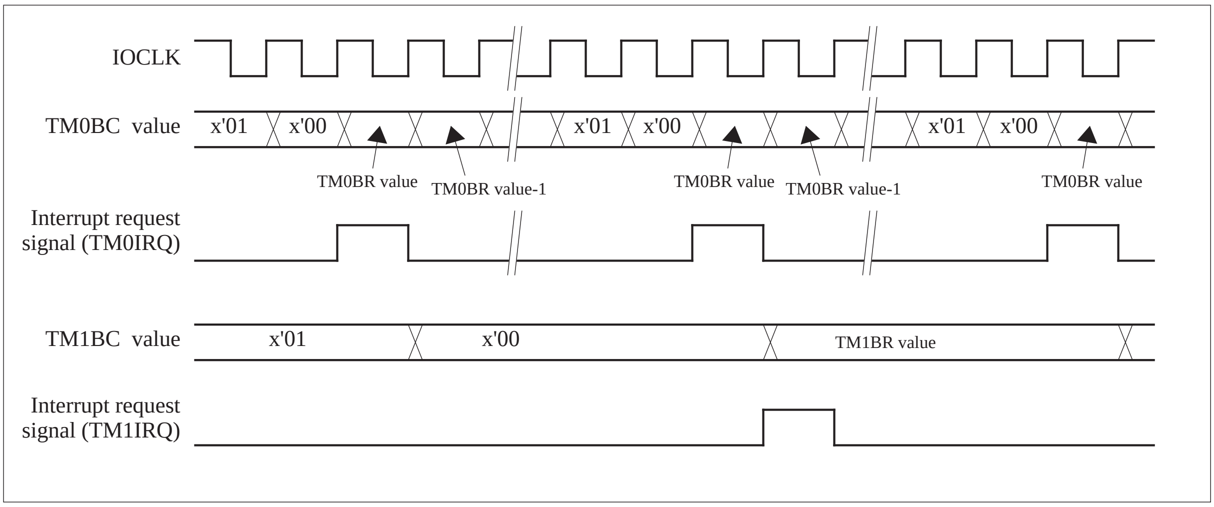


Fig. 10-6-6 Operation of Timers 0 and 1 (1)

When "cascaded with timer 0" is set, operation is as shown in Fig. 10-6-7. (IOCLK is selected as the clock source for timer 0.)
If TM1BC does not equal x'00, then when TM0BC underflows, the value in TM0BC is x'FF and the value in TM1BC is decremented by one.
If TM1BC does equal x'00, then when TM0BC underflows, the values that are set in TM0BR and TM1BR are loaded into TM0BC and TM1BC, respectively, and a timer 1 interrupt request is generated.

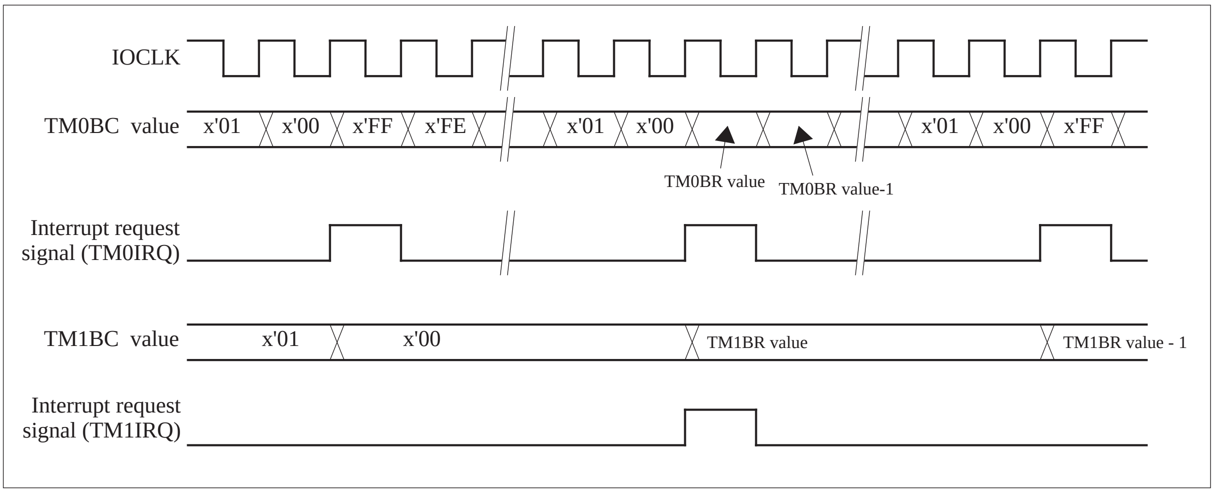


Fig. 10-6-7 Operation of Timers 0 and 1 (2)

10.6.4 PWM Output

Make the settings as described below when using an 8-bit timer to output a PWM waveform. (Timers 4 to B) The timers cannot be cascaded when outputting a PWM waveform.

■ Procedure for initiating operation

(1) Set the PWM output cycle.
Set the cycle in TMnBR.
The PWM output cycle is:
(value set in TMnBR + 1) x clock source cycle

(2) Set the PWM output duty ratio.
Set the duty ratio in TMnCMP.
The PWM output duty ratio is:
(value set in TMnCMP) / (value set in TMnBR + 1)
The set value is written to the compare register buffer.

(3) Select the clock source.
Select the clock source through TMnCK[2:0] in the TMnMD register. When using 1/8 IOCLK or 1/32 IOCLK as the clock source, set TMPSCNE to "1" in the TMPSCNT register to enable prescaler operation.

(4) Set the output mode (timers 4 to B only).
Set TMnOM[1:0] in the TMnMD register to PWM output, and select the polarity upon initialization.

(5) Initialize the timer.
Set TMnLDE in the TMnMD register to "1" to initialize timer n.
The value set in TMnBR is loaded into TMnBC as the initial value.
The value in the compare register buffer is loaded into the TMnCMP register.
PWM output is initialized.
After initialization, be certain to set TMnLDE to "0" and to return to normal operation.

(6) Set the I/O port.
Select the output signal through the TMOSL register. (Select the output for timers 8 to B.)
Set the I/O port to the timer output pin .
Select timer output for the output signal in the I/O port register, and set it to the output pin
For details on the I/O port register settings, refer to chapter 15, "I/O Ports.".

(7) Enable the timer count operation.
Set TMnCNE to "1" in the TMnMD register to start the counting operation.

Once the counting operation is enabled, the PWM waveform is output and an underflow interrupt request is generated. (Refer to Fig. 10-6-8 and 10-6-9.)
If the value in the TMnBR register is changed while the counting operation is in progress, that value is loaded as the initial value when the next underflow is generated, and the cycle of the PWM waveform changes.
If the value in the TMnCMP register is changed while the counting operation is in progress, that value is loaded into the compare register when the next underflow is generated, and the duty ratio of the PWM waveform changes.

■ Procedure for ending operation

(1) Stop the timer counting operation.
Set TMnCNE to "0" in the TMnMD register, stopping the counting operation.

(2) Initialize the timer, if necessary.
If TMnLDE is set to "1" in the TMnMD register, the timer is initialized.
The value set in TMnBR is loaded into TMnBC as the initial value.
The value in the compare register buffer is loaded into the TMnCMP register.
PWM output is initialized.
If only the timer is stopped and "1" is not written to TMnLDE, the status of the binary counter is maintained as it was before the counting operation was stopped. If TMnCNE is set to "1", the count resumes from the state that was in effect immediately before the counting operation was stopped.

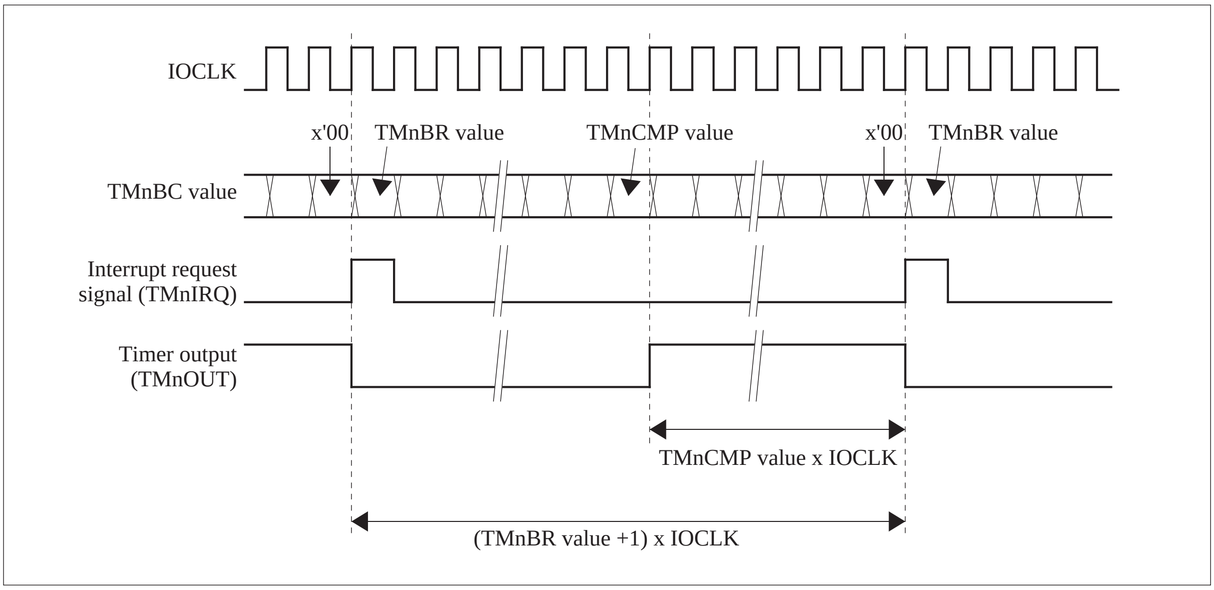


Fig. 10-6-8 PWM Output (When Clock Source = IOCLK, and "L" Level Is Output Upon Initialization)

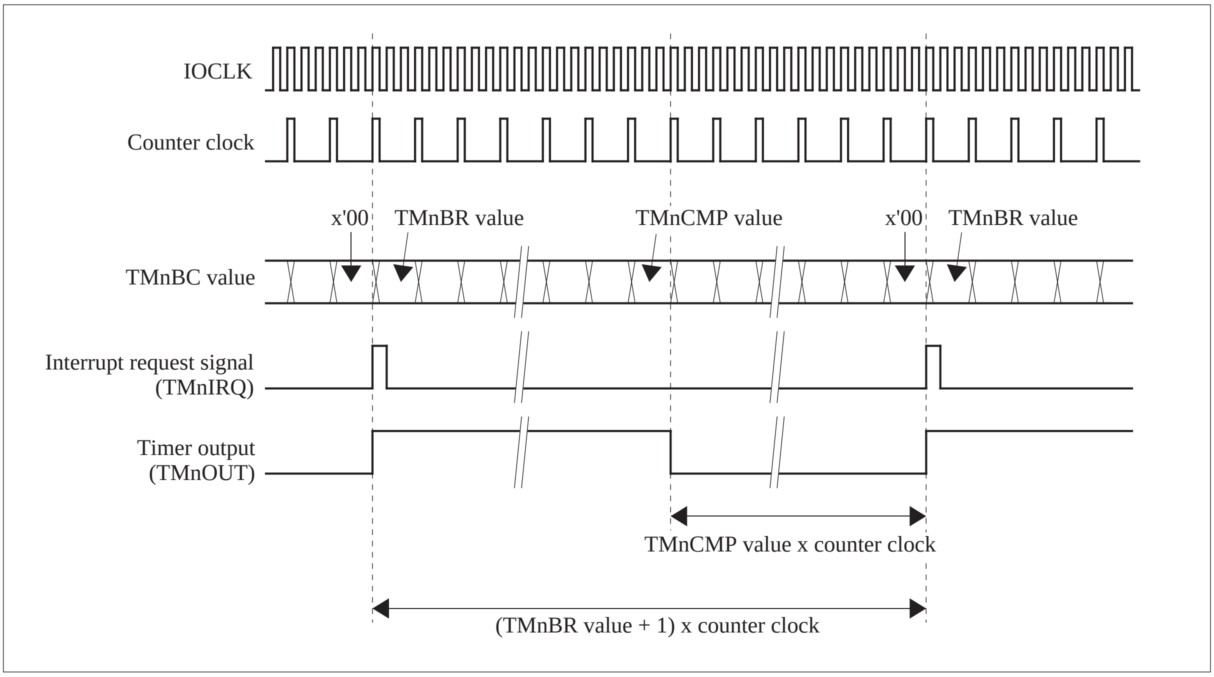

IOCLK
Counter clock
x'00
TMnBR value
TMnCMP value
x'00
TMnBR value
TMnBC value
Interrupt request signal (TMnIRQ)
Timer output (TMnOUT)
TMnCMP value x counter clock
(TMnBR value + 1) x counter clock

Fig. 10-6-9 PWM Output (When Using Prescaler, and "H" Level Is Output Upon Initialization)

11. 16-bit Timers

11.1 Overview

This microcontroller has four 16-bit timers built in.
Three are reload timers (down-counters) that can be used as interval timers or event counters.
The other is an up-counter that has two compare/capture registers built in.

11.2 Features

The features of the 16-bit timers are described below.

Timer 10

- Up-counter
- Clock sources
  An internal clock or an external clock can be selected as the clock source.
  - Internal clock: IOCLK, 1/8 IOCLK, 1/32 IOCLK, or underflow in timers 0 to 2
  - External clock: Counts the rising edge or falling edge of the input signal on the TM10IOB pin.
- Compare/capture register
  Has two compare/capture registers built in.
- Pin output
  Capable of PWM output with variable cycle and duty ratio. (One output)
  Capable of PWM output with added bits. (Two outputs)
  (Resolution: 8 + 2 bits, 8 + 3 bits, 8 + 4 bits, and 8 + 6 bits)
  Capable of one-shot output. (Two outputs)
  Polarity of pin output can be set.
- Input capture
  Each pin can be set individually to rising edge, falling edge, or both edges. (Two inputs)
  An interrupt request is generated upon capture.
  When "both edges" is set, an interrupt request is generated at both the rising edge and the falling edge.
- Interrupts
  An interrupt request is generated when the binary counter overflows.
  An interrupt request is generated when the compare register and the binary counter match, or when capture occurs. (Two outputs)

- Counting start by external trigger
  Counting can be started by input on the TM10IOB pin.
  (Edge specification possible)

Timers 11, 12, and 13

- Reload timers (down-counters)
- Clock sources
  An internal clock or an external clock can be selected as the clock source.
  - Internal clock: IOCLK, 1/8 IOCLK, 1/32 IOCLK, or underflow in timers 0 to 2
  - External clock: Counts the rising edge of the signal on the pin input.
- Interrupts
  An interrupt request is generated when the binary counter underflows.
- Timer output
  Output at 1/2 of the timer underflow is possible.

11.3 Block Diagram

Fig. 11-3-1 shows the block diagram for timer 10, and Fig. 11-3-2 shows the block diagram for timers 11 to 13.

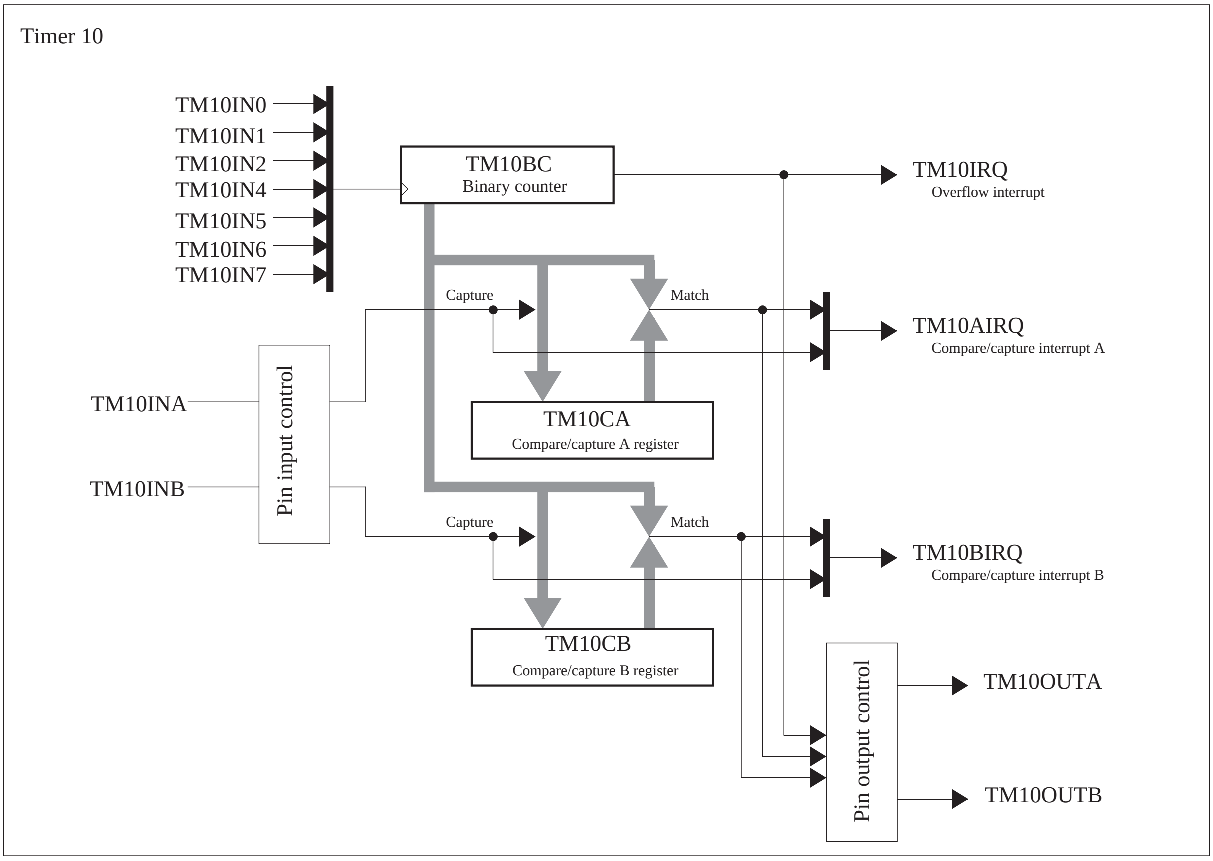


Fig. 11-3-1 16-bit Timer Block Diagram (Timer 10)

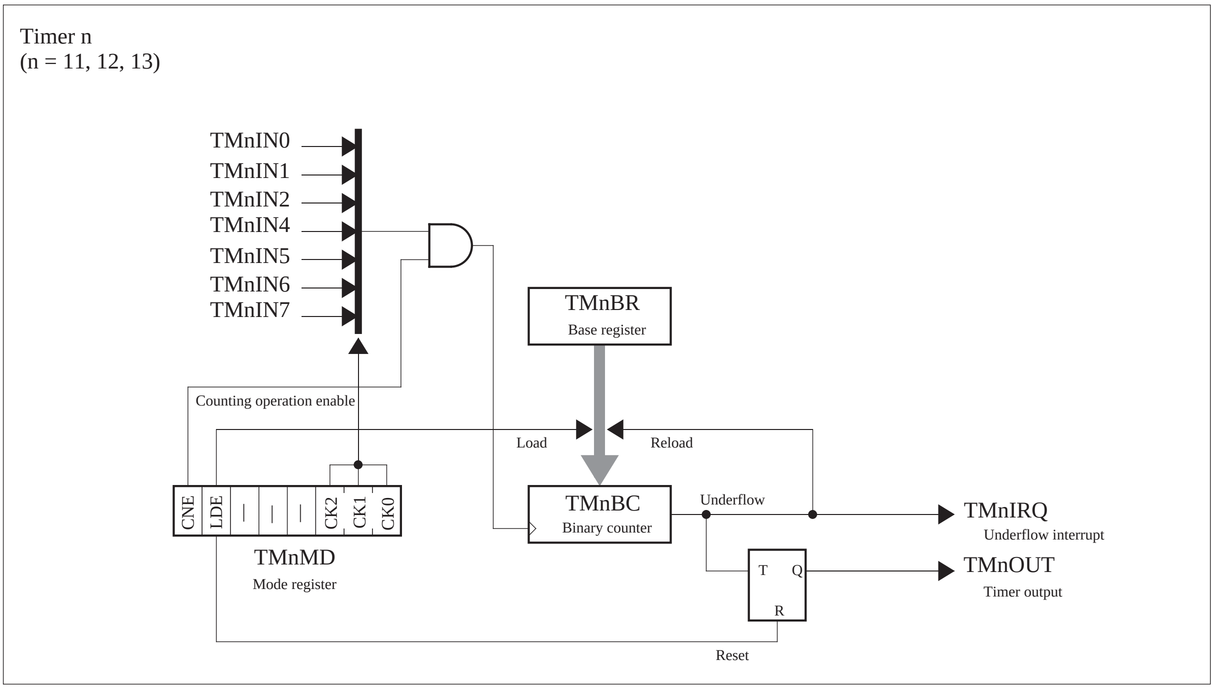


Fig. 11-3-2 16-bit Timer Block Diagram (Timers 11, 12, and 13)

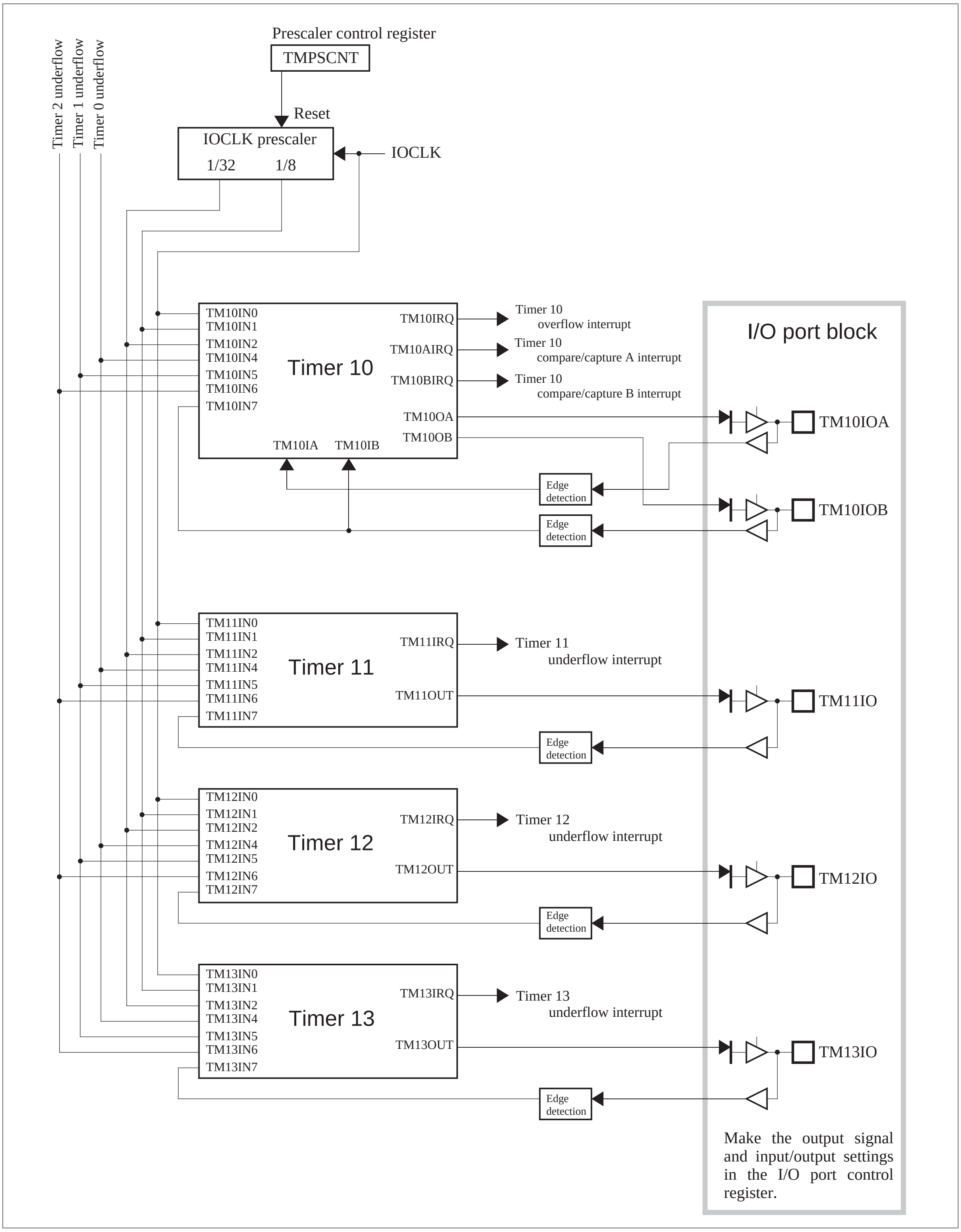


Fig. 11-3-3 16-bit Timer Connection Diagram

Fig. 11-3-4 shows the block diagram for the timer 10 compare/capture registers.

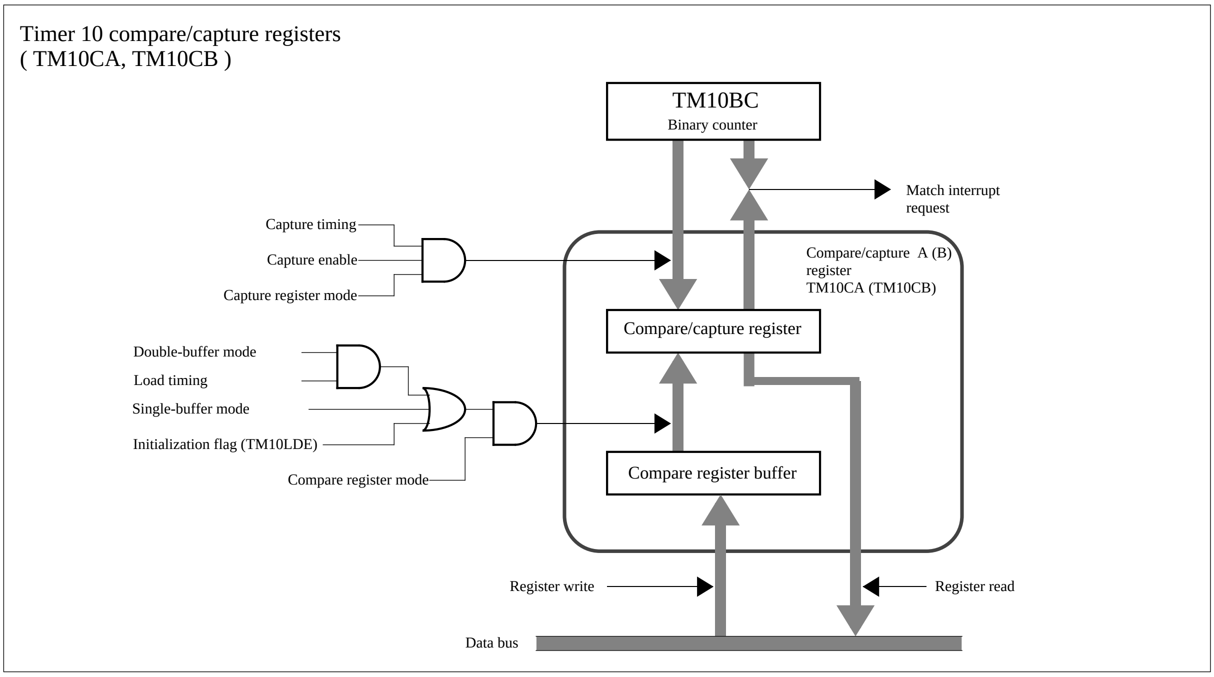


Fig. 11-3-4 Timer 10 Compare/Capture Register Block Diagram

Fig. 11-3-5 shows the block diagram for the PWM output section when timer 10 is set to PWM mode with additional bits.

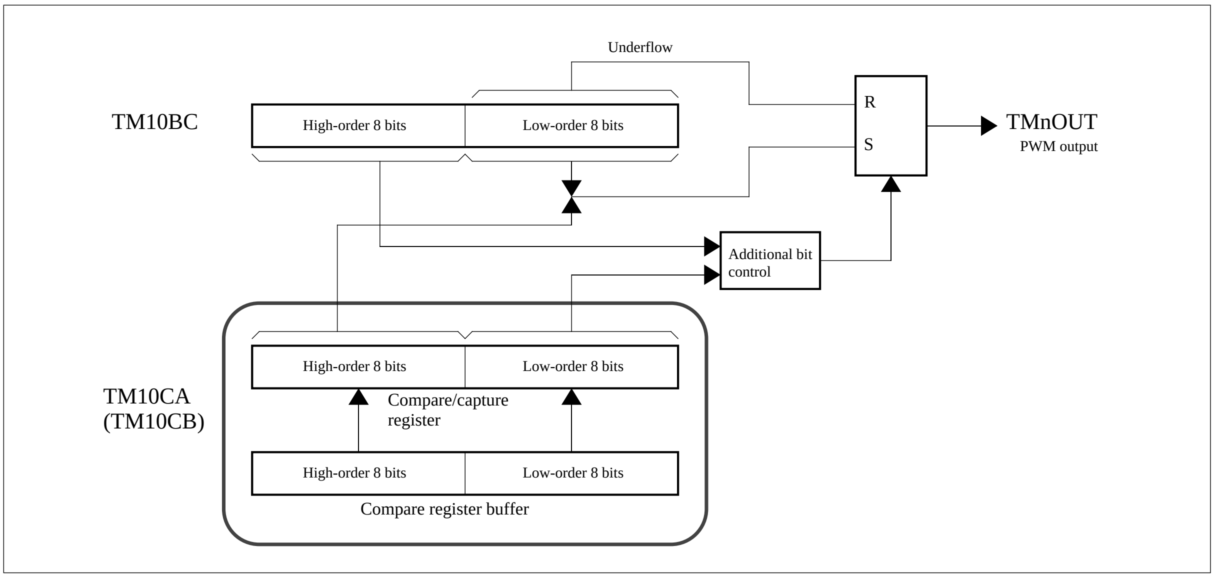


Fig. 11-3-5 PWM Output Section Block Diagram

11.4 Functions

Table 11-4-1 lists the functions of each 16-bit timer.

Table 11-4-1 List of 16-bit Timer Functions

Timer	Timer 10	Timer 11	Timer 12	Timer 13
Up-/down-counting	Up-counting	Down-counting	Down-counting	Down-counting
Interval timer	✓	✓	✓	✓
Event counting	✓ (Edge can be selected)	✓ (Rising edge)	✓ (Rising edge)	✓ (Rising edge)
Toggled output	✓ (2 outputs)	✓	✓	✓
PWM output	Variable cycle and duty ratio: 1 output Fixed cycle: 2 outputs	—	—	—
Interrupts	Overflow Compare/capture A Compare/capture B	Underflow	Underflow	Underflow
Input capture	2 inputs (either single edge or both edges)	—	—	—
One-shot output	✓	—	—	—

11.5 Description of Registers

Table 11-5-1 lists the 16-bit timer registers.

Table 11-5-1 List of 16-bit Timer Registers

Address	Name	Symbol	Number of bits	Initial value	Access size
x'34001080	Timer 10 mode register	TM10MD	16	x'0000	8, 16
x'34001082	Timer 11 mode register	TM11MD	8	x'00	8
x'34001084	Timer 12 mode register	TM12MD	8	x'00	8
x'34001086	Timer 13 mode register	TM13MD	8	x'00	8
x'34001092	Timer 11 base register	TM11BR	16	x'0000	16
x'34001094	Timer 12 base register	TM12BR	16	x'0000	16
x'34001096	Timer 13 base register	TM13BR	16	x'0000	16
x'340010A0	Timer 10 binary counter	TM10BC	16	x'0000	16
x'340010A2	Timer 11 binary counter	TM11BC	16	x'0000	16
x'340010A4	Timer 12 binary counter	TM12BC	16	x'0000	16
x'340010A6	Timer 13 binary counter	TM13BC	16	x'0000	16
x'340010B0	Timer 10 compare/capture A mode register	TM10MDA	8	x'00	8, 16
x'340010B1	Timer 10 compare/capture B mode register	TM10MDB	8	x'00	8
x'340010C0	Timer 10 compare/capture A register	TM10CA	16	x'0000	16
x'340010D0	Timer 10 compare/capture B register	TM10CB	16	x'0000	16
x'34001071	Prescaler control register	TMPSCNT	8	x'00	8

The prescaler control register (TMPSCNT) is also used by the 8-bit timers.

Timer 10 mode register

Register symbol: TM10MD
Address: x'34001080
Purpose: This register controls the operation of timer 10.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	TM10 CNE	TM10 LDE	TM10 PME	TM10 PM1	TM10 PM0	–	–	–	TM10 TGE	TM10 ONE	–	TM10 CAE	–	TM10 CK2	TM10 CK1	TM10 CK0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R	R/W	R	R/W	R/W	R/W

Bit No.	Bit name	Description
0	TM10CK0	Timer 10 clock source selection flag (LSB)
1	TM10CK1	Timer 10 clock source selection flag
2	TM10CK2	Timer 10 clock source selection flag (MSB) These bits select the clock source for timer 10. When pin input is selected, counting occurs at the rising edge of the pin input. Note : For details on the clock sources for each timer, refer to Table 11-5-2, "16-bit Timer Clock Sources."
3	—	"0" is returned when this bit is read.
4	TM10CAE	Counter clear enable flag Enables/disables clearing of TM10BC when TM10BC and TM10CA match. 0: Do not clear. (TM10BC becomes a 16-bit free-running counter.) 1: Clear. When TM10CA is set as a compare register; TM10BC is cleared if TM10BC and TM10CA match. When TM10CA is set as a capture register; TM10BC is cleared when a capture occurs in TM10CA.
5	—	"0" is returned when this bit is read.
6	TM10ONE	One-shot operation enable flag Enables/disables the halt of timer operation when TM10BC and TM10CA match. 0: Disables one-shot operation. 1: Enables one-shot operation. If TM10BC and TM10CA match, the TM10CNE flag is reset and the timer stops.

Bit No.	Bit name	Description
7	TM10TGE	External trigger start enable flag Enables/disables timer start by an external trigger. 0: Disables timer start by an external trigger. (The trigger input is ignored.) 1: Enables timer start by an external trigger. When the specified edge is input to the TM10IOB pin, the TM10CNE flag is set and the timer starts. The timer starts on the edge that is the opposite of the one selected by the TM10BEG flag in the TM10MDB register.
10 to 8	—	"0" is returned when these bits are read.
11	TM10PM0	Timer 10 PWM output resolution selection flag (LSB)
12	TM10PM1	Timer 10 PWM output resolution selection flag (MSB) These bits select the resolution for PWM output with additional bits for timer 10. 00: 10 bits (basic wave 8 bits + 2 bits) 01: 11 bits (basic wave 8 bits + 3 bits) 10: 12 bits (basic wave 8 bits + 4 bits) 11: 14 bits (basic wave 8 bits + 6 bits)
13	TM10PME	Timer 10 PWM output waveform selection flag This bit selects the PWM output waveform for timer 10. 0: Normal waveform 1: PWM output with additional bits The PWM waveform is output with the resolution that is set in TM10PM1 and 0. The number of bits that was set in TM10PM1 and 0 is the number of bits in TM10BC that function as a binary counter.
14	TM10LDE	Timer 10 initialization flag Initializes timer 10. 0: Normal operation 1: Initialize Clears TM10BC (so that TM10BC = x'0000). When TM10CA and TM10CB are set as a double-buffer compare register, the value in the buffers is loaded into the compare register. Also initializes the pin output.
15	TM10CNE	Timer 10 operation enable flag Enables/disables the timer 10 counting operation. 0: Operation disabled. 1: Operation enabled.

[Note]
When setting TM10CNE to "1", do so while TM10LDE is set to "0".
When setting TM10LDE to "1", do so while TM10CNE is set to "0".
Operation is not guaranteed if TM10CNE and TM10LDE are both set to "1" at the same time.

Timer n mode register (n = 11, 12, 13)

Register symbol: TMnMD

Address: x'34001082 (n=11), x'34001084 (n=12), x'34001086 (n=13)

Purpose: This register controls the operation of timer n.

Bit No.	7	6	5	4	3	2	1	0
Bit name	TMn CNE	TMn LDE	–	–	–	TMn CK2	TMn CK1	TMn CK0
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R	R	R	R/W	R/W	R/W

Bit No.	Bit name	Description
0	TMnCK0	Timer n clock source selection flag (LSB)
1	TMnCK1	Timer n clock source selection flag
2	TMnCK2	Timer n clock source selection flag (MSB) These bits select the clock source for the timers. When pin input is selected, counting occurs at the rising edge of the pin input. For details on the clock sources for each timer, refer to Table 11-5-2, "16-bit Timer Clock Sources".
5 to 3	—	"0" is returned when these bits are read.
6	TMnLDE	Timer n initialization flag Initializes timer n. 0: Normal operation 1: Initialize Loads the value in TMnBR into TMnBC, resets timer output n, and loads the value in the compare register buffer into the compare register.
7	TMnCNE	Timer n operation enable flag Enables/disables the timer n counting operation. 0: Operation disabled. 1: Operation enabled.

[Note]
When setting TMnCNE to "1", do so while TMnLDE is set to "0".
When setting TMnLDE to "1", do so while TMnCNE is set to "0".
Operation is not guaranteed if TMnCNE and TMnLDE are both set to "1" at the same time.

Table 11-5-2 16-bit Timer Clock Sources

TMnCK [2:0] Setting value	Timer 10	Timer 11	Timer 12	Timer 13
000	IOCLK	IOCLK	IOCLK	IOCLK
001	IOCLK/8	IOCLK/8	IOCLK/8	IOCLK/8
010	IOCLK/32	IOCLK/32	IOCLK/32	IOCLK/32
011	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
100	Timer 0 underflow	Timer 0 underflow	Timer 0 underflow	Timer 0 underflow
101	Timer 1 underflow	Timer 1 underflow	Timer 1 underflow	Timer 1 underflow
110	Timer 2 underflow	Timer 2 underflow	Timer 2 underflow	Timer 2 underflow
111	TM10IOB pin input	TM11IO pin input	TM12IO pin input	TM13IO pin input

Timer n base register (n = 11, 12, 13)

Register symbol: TMnBR

Address: x'34001092 (n=11), x'34001094 (n=12), x'34001096 (n=13)

Purpose: This register sets the initial value and the underflow cycle for the timer n binary counter.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	TMn BR15	TMn BR14	TMn BR13	TMn BR12	TMn BR11	TMn BR10	TMn BR9	TMn BR8	TMn BR7	TMn BR6	TMn BR5	TMn BR4	TMn BR3	TMn BR2	TMn BR1	TMn BR0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The value set in TMnBR is loaded into TMnBC under the following conditions:

(1) When TMnLDE = 1

(2) When an underflow has occurred.

TMnBC generates an underflow interrupt request every (value set in TMnBR + 1) counts.

Timer n binary counter (n = 10, 11, 12, 13)

Register symbol: TMnBC

Address: x'340010A0 (n=10), x'340010A2 (n=11),
x'340010A4 (n=12), x'340010A6 (n=13)

Purpose: This is the binary counter for timer n.
The counter value can be read from this register.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	TMn BC15	TMn BC14	TMn BC13	TMn BC12	TMn BC11	TMn BC10	TMn BC9	TMn BC8	TMn BC7	TMn BC6	TMn BC5	TMn BC4	TMn BC3	TMn BC2	TMn BC1	TMn BC0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Timer 10 is an up-counter that counts up from an initial value of x'0000, and generates an interrupt request when an overflow occurs. In PWM output mode with additional bits, timer 10 operates as a binary counter with the resolution that was set, and generates an interrupt request when an overflow occurs.

Timers 11, 12 and 13 are down-counters. With the value set in TMnBR as the initial value, these registers underflow after (value set in TMnBR + 1) counts, and generate an interrupt request.

Timer 10 compare/capture A mode register

Register symbol: TM10MDA
Address: x'340010B0
Purpose: This register controls the operation of the timer 10 compare/capture A register.
This register also sets the waveform that is output to the TM10IOA pin.

Bit No.	7	6	5	4	3	2	1	0
Bit name	TM10 AM1	TM10 AM0	TM10 AEG	TM10 ACE	–	TM10 AO2	TM10 AO1	TM10 AO0
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit No.	Bit name	Description
0	TM10AO0	Timer 10A output waveform selection flag (LSB)
1	TM10AO1	Timer 10A output waveform selection flag
2	TM10AO2	Timer 10A output waveform selection flag (MSB) These bits select the waveform that is output to the TM10IOA pin. 000: Set when TMnBC matches TMnCA, reset when TMnBC matches TMnCB. 001: Set when TMnBC matches TMnCA, reset when TMnBC overflows. 010: Set when TMnBC matches TMnCA, reset only when the timer is initialized. 011: Reset when TMnBC matches TMnCA. 100: Toggled output (Output is inverted when TMnBC matches TMnCA.) 101, 110, 111: Setting prohibited.
3	—	"0" is returned when this bit is read.
4	TM10ACE	Timer 10 capture A operation enable flag Enables/disables capture operation for TM10CA. 0: Disables capture operation. (Pin input is ignored.) 1: Enables capture operation.
5	TM10AEG	Timer 10 A pin polarity selection flag Selects the valid edge for the input on the TM10IOA pin, and the output polarity. 0: Rising edge valid Positive polarity output (Reset: “L” level; set: “H” level) 1: Falling edge valid Negative polarity output (Reset: “H” level; set: “L” level)
6	TM10AM0	Timer 10 compare/capture A operating mode setting flag (LSB)
7	TM10AM1	Timer 10 compare/capture A operating mode setting flag (MSB) These bits set the TM10CA operating mode. 00: Compare register (single-buffer) 01: Compare register (double-buffer) 10: Capture register (single-edge operation) 11: Capture register (dual-edge operation)

When dual-edge capture is set, the setting of TM10AEG is ignored.

Timer 10 compare/capture B mode register

Register symbol: TM10MDB
Address: x'340010B1
Purpose: This register controls the operation of the timer 10 compare/capture B register.
This register also sets the waveform that is output to TM10IOB pin.

Bit No.	7	6	5	4	3	2	1	0
Bit name	TM10 BM1	TM10 BM0	TM10 BEG	TM10 BCE	–	TM10 BO2	TM10 BO1	TM10 BO0
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit No.	Bit name	Description
0	TM10BO0	Timer 10B output waveform selection flag (LSB)
1	TM10BO1	Timer 10B output waveform selection flag
2	TM10BO2	Timer 10B output waveform selection flag (MSB) These bits select the waveform that is output to the TM10IOB pin. 000: Set when TMnBC matches TMnCB, reset when TMnBC matches TMnCA. 001: Set when TMnBC matches TMnCB, reset when TMnBC overflows. 010: Set when TMnBC matches TMnCB, reset only when the timer is initialized. 011: Reset when TMnBC matches TMnCB. 100: Toggled output (Output is inverted when TMnBC matches TMnCB.) 101, 110, 111: Setting prohibited.
3	—	"0" is returned when this bit is read.
4	TM10BCE	Timer 10 capture B operation enable flag Enables/disables capture operation for TM10CB. 0: Disables capture operation. (Pin input is ignored.) 1: Enables capture operation.
5	TM10BEG	Timer 10B pin polarity selection flag Selects the valid edge for the input on the TM10IOB pin, and the output polarity. 0: Rising edge valid Positive polarity output (Reset: “L” level; set: “H” level) 1: Falling edge valid Negative polarity output (Reset: “H” level; set: “L” level)
6	TM10BM0	Timer 10 compare/capture B operating mode setting flag (LSB)
7	TM10BM1	Timer 10 compare/capture B operating mode setting flag (MSB) These bits set the TM10CB operating mode. 00: Compare register (single-buffer) 01: Compare register (double-buffer) 10: Capture register (single-edge operation) 11: Capture register (dual-edge operation)

When dual-edge capture is set, the setting of TM10BEG is ignored.

Timer 10 compare/capture A register

Register symbol: TM10CA
Address: x'340010C0
Purpose: This is the timer 10 compare/capture A register.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	TM10 CA15	TM10 CA14	TM10 CA13	TM10 CA12	TM10 CA11	TM10 CA10	TM10 CA9	TM10 CA8	TM10 CA7	TM10 CA6	TM10 CA5	TM10 CA4	TM10 CA3	TM10 CA2	TM10 CA1	TM10 CA0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

When this register is set as a compare register, an interrupt request is generated when TM10BC and TM10CA match.
The timer 10 cycle can be set by clearing TM10BC when TM10BC matches TM10CA.
The cycle is the set value + 1.
When this register is set as a double-buffer compare register, data that is written to TM10CA is stored temporarily in a buffer, so it is possible that after writing TM10CA, a read of TM10CA will still return the value that was previously stored there.
The value set in the buffer is loaded into the compare register under the conditions described below. In any of these cases, the value in TM10BC becomes x'0000.

(1) When timer 10 is initialized
(2) When an overflow occurs (while TM10CAE is set to "0")
(3) When TM10BC matches TM10CA (while TM10CAE is set to "1")

When this register is set as a capture register, the value in TM10BC is captured in TM10CA and an interrupt request is generated when the edge that was selected by the TM10AEG flag is input to the TM10IOA pin.
When this register is set as a dual-edge capture register, the value in TM10BC is captured in TM10CA and an interrupt request is generated at either a rising edge or a falling edge.

Timer 10 compare/capture B register

Register symbol: TM10CB
Address: x'340010D0
Purpose: This is the timer 10 compare/capture B register.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	TM10 CB15	TM10 CB14	TM10 CB13	TM10 CB12	TM10 CB11	TM10 CB10	TM10 CB9	TM10 CB8	TM10 CB7	TM10 CB6	TM10 CB5	TM10 CB4	TM10 CB3	TM10 CB2	TM10 CB1	TM10 CB0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

When this register is set as a compare register, an interrupt request is generated when TM10BC and TM10CB match.
The timer 10 cycle can be set by clearing TM10BC when TM10BC matches TM10CB.
The cycle is the set value + 1.
When this register is set as a double-buffer compare register, data that is written to TM10CB is stored temporarily in a buffer, so it is possible that after writing TM10CB, a read of TM10CB will still return the value that was previously stored there.
The value set in the buffer is loaded into the compare register under the conditions described below. In any of these cases, the value in TM10BC becomes x'0000.

(1) When timer 10 is initialized
(2) When an overflow occurs (while TM10CAE is set to "0")
(3) When TM10BC matches TM10CA (while TM10CA is set as a compare register, and TM10CAE is set to "1".)
(4) When capturing a value in TM10CA (while TM10CA is set as a capture register, and TM10CAE is set to "1".)

When this register is set as a capture register, the value in TM10BC is captured in TM10CB and an interrupt request is generated when the edge that was selected by the TM10BEG flag is input to the TM10IOB pin.
When this register is set as a dual-edge capture register, the value in TM10BC is captured in TM10CB and an interrupt request is generated at either a rising edge or a falling edge.

Prescaler control register

Register symbol: TMPSCNT
Address: x'34001071
Purpose: This register controls prescaler operations.

Bit No.	7	6	5	4	3	2	1	0
Bit name	TMPS CNE	–	–	–	–	–	–	–
Reset	0	0	0	0	0	0	0	0
Access	R/W	R	R	R	R	R	R	R

Bit No.	Bit name	Description
6 to 0	—	"0" is returned when these bits are read.
7	TMPSCNE	Prescaler operation enable flag Enables/disables 1/8 IOCLK and 1/32 IOCLK prescaler operation. 0: Prescaler operation disable 1: Prescaler operation enabled.

This prescaler also serves as the 1/8 IOCLK or 1/32 IOCLK prescaler that is used by 8-bit timers.

11.6 Description of Operation of Timer 10

This section describes the operation of timer 10.
Timer 10 includes an up-counter and two compare/capture registers. The compare/capture registers are independent of each other, and can each be used as either a compare register or a capture register.

11.6.1 Compare Register Settings

In order to use either the timer 10 compare/capture A register or B register as a compare register, the following settings must be made according to the procedure described below before timer 10 is initialized.
The explanation below refers only to the compare/capture A register, but similar settings would also need to be made for the compare/capture B register.

(1) Set the compare/capture A register mode.
Set the TM10MDA register as follows:

TM10AO2,1,0	Don't care
TM10ACE	0: Capture operation disabled
TM10AEG	Don't care
TM10AM1,0	00: Compare register (single-buffer) or 01: Compare register (double-buffer) If the value in the compare register will change while the counting operation is in progress, be certain to set "double-buffer."

(2) Set the comparison value in the compare/capture A register.
Set the comparison value in TM10CA.
If the double-buffer is set, the value that is set is not loaded into the compare register at this point. Even if TM10CA is read at this point, the previous value that was set is read. The set value is loaded when timer 10 is initialized.

Once the timer 10 counting operation is enabled, a compare/capture A interrupt request is generated according to the timing shown in Fig. 11-6-1.
If the double-buffer is set, the value that is set in the buffer is loaded into the compare register at the same time that TM10BC is cleared.

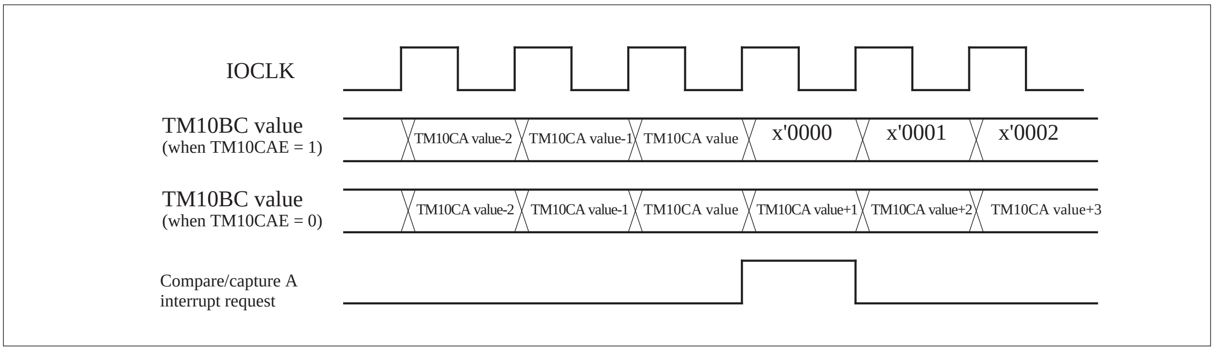


Fig. 11-6-1 Compare Register Operation (When Clock Source = IOCLK)

11.6.2 Capture Register Settings

In order to use either the timer 10 compare/capture A register or B register as a capture register, the following settings must be made according to the procedure described below before timer 10 is initialized.
The explanation below refers only to the compare/capture A register, but similar settings would also need to be made for the compare/capture B register.

(1) Set the compare/capture A register mode.
Set the TM10MDA register as follows:
TM10AO2,1,0 Don't care
TM10ACE 1: Capture operation enabled
TM10AEG Don't care (If "single-edge" is selected below, select either rising edge or falling edge here.)
TM10AM1,0 10: Capture register (single-edge)
or
11: Capture register (dual-edge)
If "dual-edge" is selected, the setting in TM10AEG is ignored.

Once the timer 10 counting operation is enabled, the value in TM10BC is captured in TM10CA, and a compare/capture A interrupt request is generated according to the timing shown in Fig. 11-6-2. (If the counting operation is currently halted, the capture operation does not occur even if the selected edge is input to the pin.)

If dual-edge was selected, the capture operation is performed when either a rising or falling edge is input. It is not possible to determine which edge was input. (The pin input level cannot be read.)
The capture operation can be disabled even while counting is in progress by setting TM10ACE to "0".

When TM10CAE is set to "1" in the TM10MD register and TM10CA is set as a capture register, TM10BC is cleared when the value is captured in TM10CA. If TM10CB is set as a double-buffer compare register, the value that is set in the buffer is loaded into the compare register at this time.

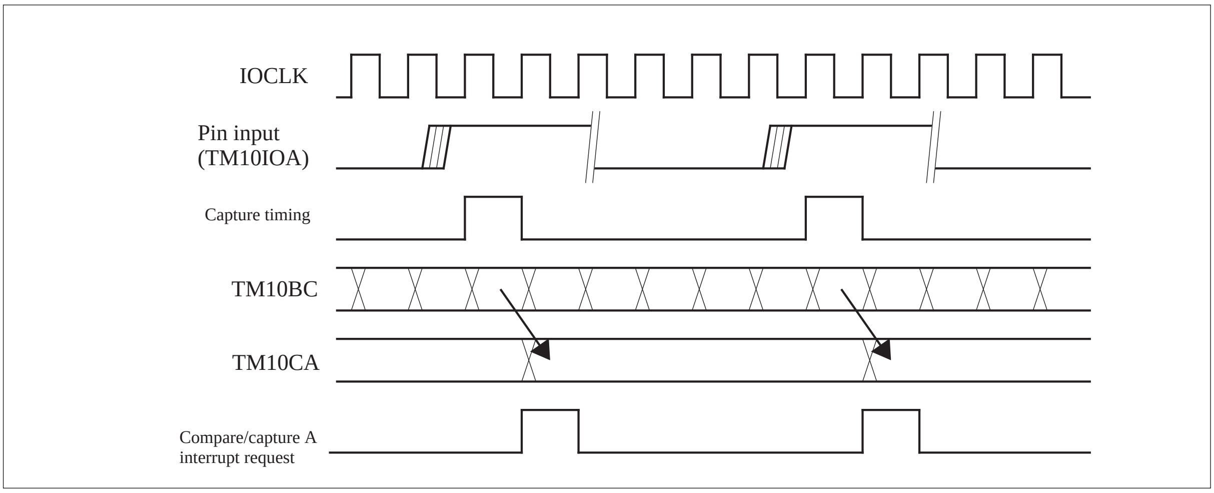


Fig. 11-6-2 Input Capture Operation (When "Rising Edge" Is Selected)

11.6.3 Pin Output Settings

Timer 10 can be used to output a variety of waveforms to the TM10IOA and TM10IOB pins.

(1) Setting the output level upon initialization
If the TM10LDE flag in the TM10MD register is set to "1", thus initializing timer 10, the output level on the TM10IOA pin is the value that is set for the TM10AEG flag in the TM10MDA register.
The output level on the TM10IOB pin is the value that is set for the TM10BEG flag in the TM10MDB register.
Once the TM10LDE flag in the TM10MD register is returned to "0" in order to resume normal operation, manipulating the TM10AEG or TM10BEG flags does not change the output level on the corresponding pins.

(2) Setting the output waveform for the counting operation
If the TM10CNE flag in the TM10MD register is set to "1", thus enabling the timer 10 counting operation, the waveform selected by the TM10AO0, 1, and 2 flags in the TM10MDA register is output to the TM10IOA pin.
Similarly, the waveform selected by the TM10BO0, 1, and 2 flags in the TM10MDB register is output to the TM10IOB pin.
Because the values of the TM10AEG and TM10BEG flags are referenced only when the output level is changed, changing the settings of these flags does not change the output level until the next time that the output is changed.

Examples of TM10IOA pin output waveforms are shown below. Output for the TM10IOB pin is similar.
Fig. 11-6-3 shows the output waveform for the TM10IOA pin when "Set when TM10BC matches TM10CA, and reset when TM10BC matches TM10CB" is set. If the set and reset conditions occur simultaneously, the reset takes precedence.

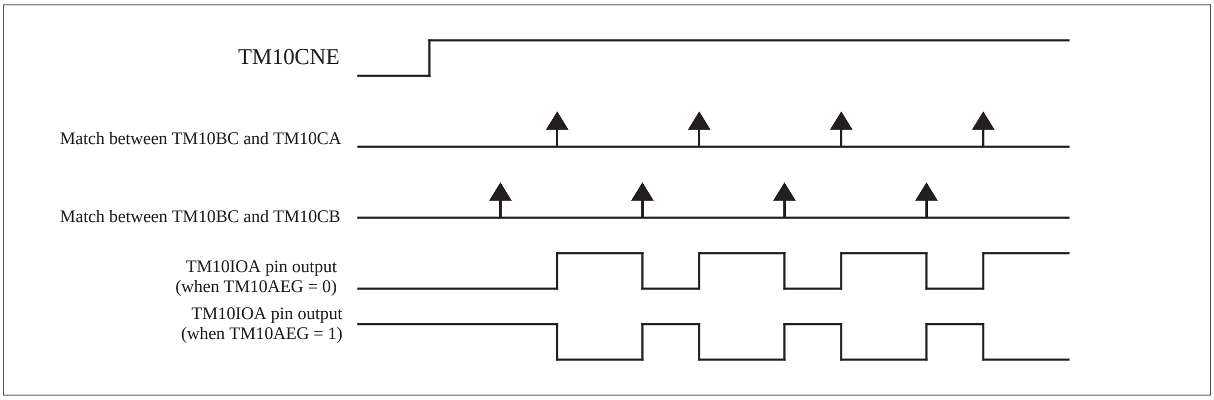


Fig. 11-6-3 Pin Output Waveform (1)

Fig. 11-6-4 shows the output waveform for the TM10IOA pin when "Set when TM10BC matches TM10CA, and reset when TM10BC overflows" is set. If the set and reset conditions occur simultaneously, the reset takes precedence.

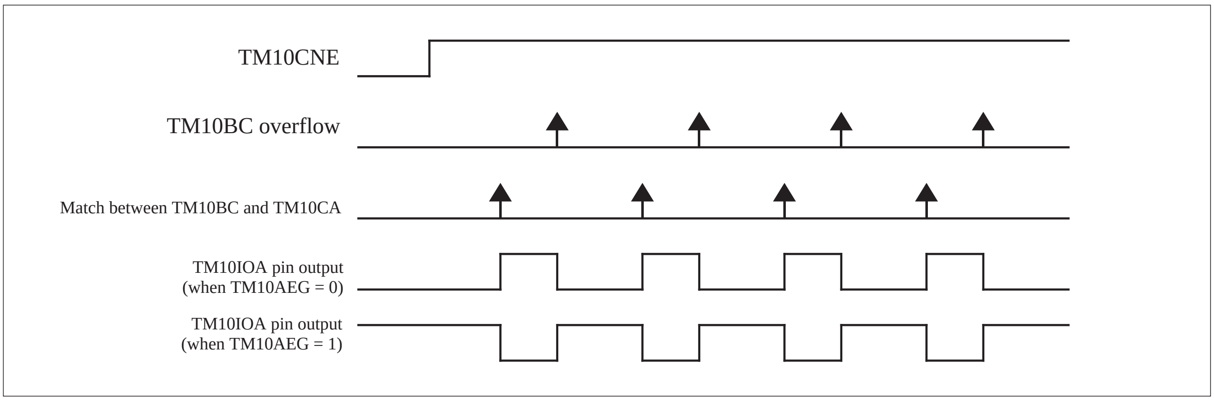


Fig. 11-6-4 Pin Output Waveform (2)

Fig. 11-6-5 shows the output waveform for the TM10IOA pin when "Set when TM10BC matches TM10CA" is set.

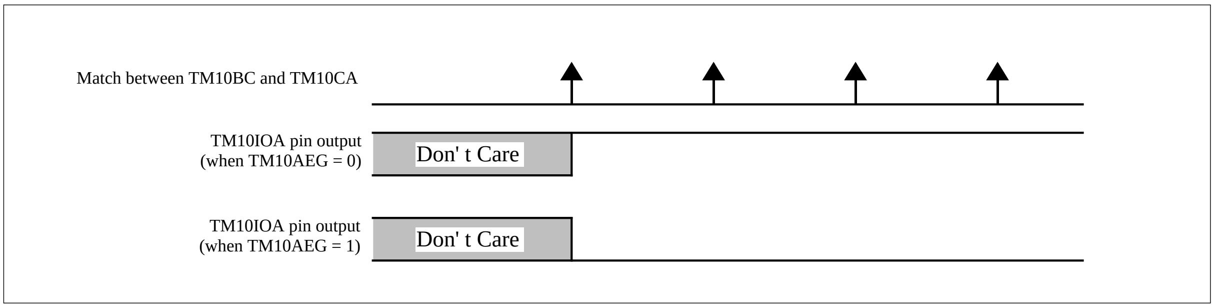

Match between TM10BC and TM10CA
TM10IOA pin output (when TM10AEG = 0)
Don' t Care
TM10IOA pin output (when TM10AEG = 1)
Don' t Care

Fig. 11-6-5 Pin Output Waveform (3)

Fig. 11-6-6 shows the output waveform for the TM10IOA pin when "Reset when TM10BC matches TM10CA" is set.

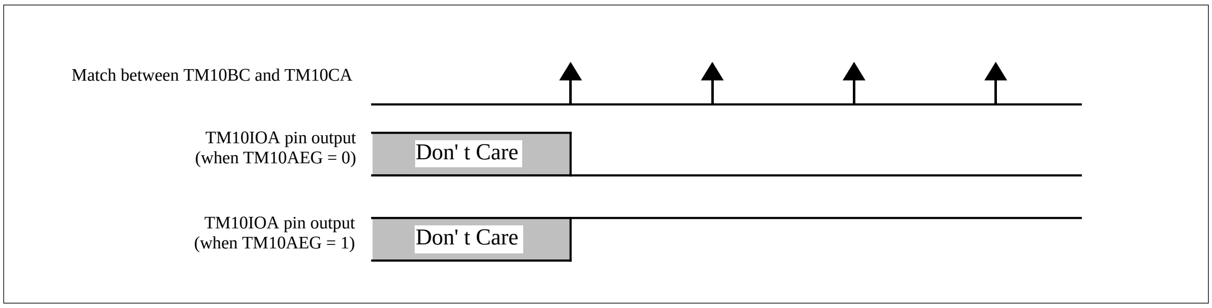

Match between TM10BC and TM10CA
TM10IOA pin output (when TM10AEG = 0)
Don' t Care
TM10IOA pin output (when TM10AEG = 1)
Don' t Care

Fig. 11-6-6 Pin Output Waveform (4)

Fig. 11-6-7 shows the output waveform for the TM10IOA pin when "Toggled output" is set.

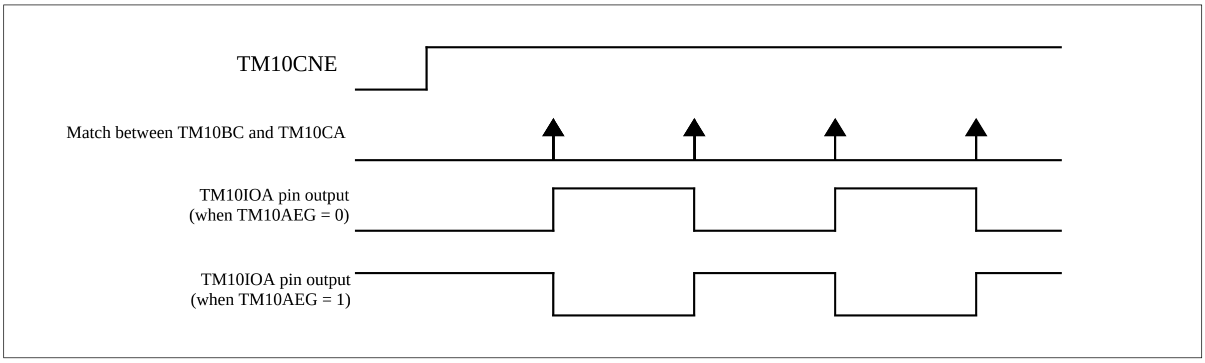

TM10CNE
Match between TM10BC and TM10CA
TM10IOA pin output (when TM10AEG = 0)
TM10IOA pin output (when TM10AEG = 1)

Fig. 11-6-7 Pin Output Waveform (5)

11.6.4 Starting by an External Trigger

Timer 10 can be started up by input on the TM10IOB pin. Fig. 11-6-8 illustrates the startup operation.
The compare/capture A and B registers can be used as compare registers or as capture registers.

■ Procedure for initiating operation

(1) Select the input edge on which timer 10 is to start.
Select the input edge through TM10BEG in the TM10MDB register.
The starting edge is the opposite of the normal setting.

TM10BEG = 0	Start when falling edge is input
TM10BEG = 1	Start when rising edge is input

(2) Set the operating mode.
Set the TM10MD register as described below:

TM10CK2,1,0	Don't care;	select any clock source.
TM10CAE	Don't care;	"1" when setting one-shot operation and the interrupt cycle.
TM10ONE	Don't care;	"1" for one-shot operation.
TM10TGE	0:	This disables timer start by an external trigger.
TM10PM1,0	Don't care;	this setting is ignored.
TM10PME	0:	Selects the normal waveform.
TM10LDE	0:	Normal operation.
TM10CNE	0:	Stops counting operation.

When using 1/8IOCLK or 1/32IOCLK as the clock source, set TMPSCNE in the TMPSCNT register to "1" to enable prescaler operation before enabling the counting operation for timer 10.

(3) Initialize the timer.
Set TM10LDE in the TM10MD register to "1" in order to initialize timer 10.
TM10BC is cleared, and the pin output is reset.
In addition, if the compare/capture register is set as a double-buffer compare register, the value in the buffer is loaded into the compare register.
After initialization is completed, be certain to reset TM10LDE back to "0" in order to restore normal operation mode.

(4) Set the I/O port.
Set TM10IOB pin to "input pin."
Set the TM10IOA pin as desired.
Note: For details on the I/O port register settings, refer to chapter 15, "I/O Ports."

(5) Enable timer startup by an external trigger.
Set TM10TGE in the TM10MD register to "1".

Once the specified edge is input to the TM10IOB pin, timer 10 starts up. (The TM10CNE flag in the TM10MD register is set by the hardware.)

■ Procedure for ending operation

(1) Disable timer startup by an external trigger.
Set TM10TGE in the TM10MD register to "0".

(2) Stop the counting operation.
Set TM10CNE in the TM10MD register to "0".

If TM10TGE and TM10CNE are both set to "0" simultaneously, there is a possibility that TM10CNE will be set again, depending on the pin input timing. Therefore, always be sure to set TM10TGE to "0" first, and then set TM10CNE to "0".

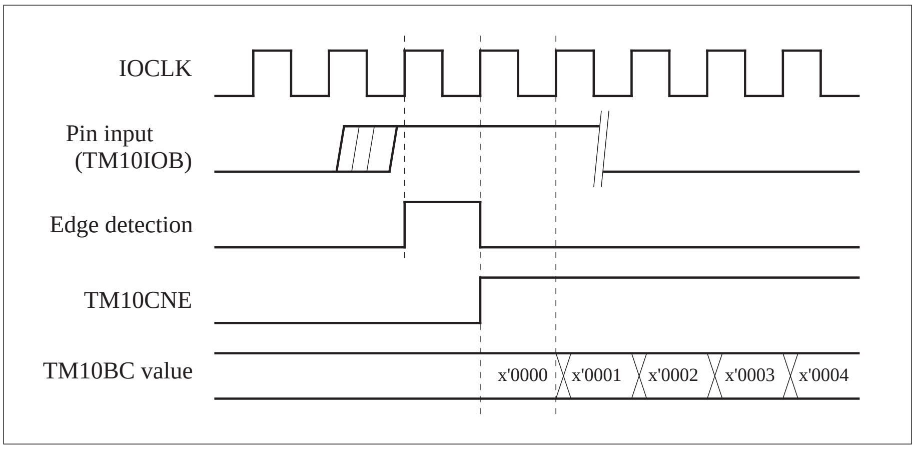


Fig. 11-6-8 Timer 10 Startup by an External Trigger (When “Rising Edge” is Selected)

11.6.5 One-shot Operation

It is possible to stop timer 10 when TM10BC and TM10CA match. Figs. 11-6-9 and 11-6-10 illustrate the operation that stops timer 10.
The compare/capture B register can be used as a compare register or as a capture register.

■ Procedure for initiating operation

(1) Set the compare/capture A register mode.
Set the TM10MDA register as follows:

TM10AO2,1,0	Don't care
TM10ACE	0: Capture operation disabled
TM10AEG	Don't care
TM10AM1,0	00: Compare register (single-buffer)
	or
	01: Compare register (double-buffer)
	If the value in the compare/capture A register will change while the counting operation is in progress, be certain to set "double-buffer."

(2) Set the value at which the timer is to stop.
Set the comparison value in TM10CA.
The timer will stop when it reaches the count of (value set in the TM10CA register + 1).

(3) Set the operating mode.
Set the TM10MD register as described below:

TM10CK2,1,0	Don't care;	Select any clock source.
TM10CAE	1:	Clears TM10BC when TM10CA matches TM10BC.
TM10ONE	1:	Enables one-shot operation.
TM10TGE	0:	Disables timer start by an external trigger.
TM10PM1,0	Don't care;	This setting is ignored.
TM10PME	0:	Selects the normal waveform.
TM10LDE	0:	Normal operation.
TM10CNE	0:	Stops counting operation.

When using 1/8 IOCLK or 1/32 IOCLK as the clock source, set TMPSCNE in the TMPSCNT register to "1" to enable prescaler operation before enabling the counting operation for timer 10.

(4) Initialize the timer.
Set TM10LDE in the TM10MD register to "1" in order to initialize timer 10.
TM10BC is cleared, and the pin output is reset.
In addition, if TM10CA is set as a double-buffer compare register, the value in the buffer is loaded into the compare register.
After initialization is completed, be certain to reset TM10LDE back to "0" in order to restore normal operation mode.

(5) Set the I/O port (when using pin output).
Set the I/O port to "timer output pin."
In the I/O port register, select "timer output" for the output signal and then set the output pin.
When the timer is to be started up by an external trigger, set the TM10IOB pin to "input pin".

(6) Enable the timer counting operation.
The counting operation starts when the TM10CNE in the TM10MD register is set to "1".
If the timer is to be started by an external trigger, leave TM10CNE set to "0" and set TM10TGE to "1".

When starting to count up again after TM10BC and TM10CA have matched, the hardware clears the TM10CNE flag to stop the counting operation. TM10BC is also cleared.

Note: For details on the I/O port register settings, refer to chapter 15, "I/O Ports."

■ Procedure for ending operation

- When the timer was started by a program (TM10TGE = 0)
  (1) Stop the counting operation.
  Set TM10CNE in the TM10MD register to "0".

- When the timer was started by an external trigger (TM10TGE = 1)
  (1) Disable timer startup by an external trigger.
  Set TM10TGE in the TM10MD register to "0".
  (2) Stop the counting operation.
  Set TM10CNE in the TM10MD register to "0".

If TM10TGE and TM10CNE are both set to "0" simultaneously, there is a possibility that TM10CNE will be set again, depending on the pin input timing. Therefore, always be sure to set TM10TGE to "0" first, and then set TM10CNE to "0".

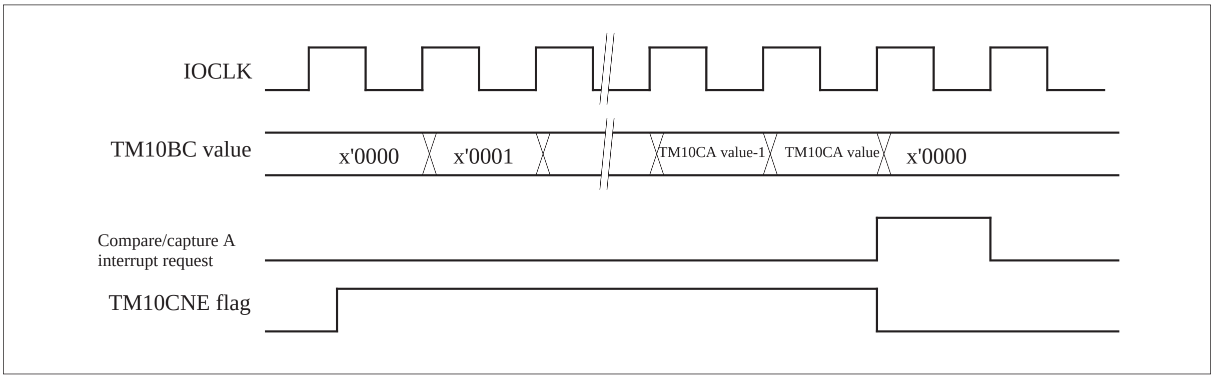


Fig. 11-6-9 One-shot Operation (When Clock Source = IOCLK)

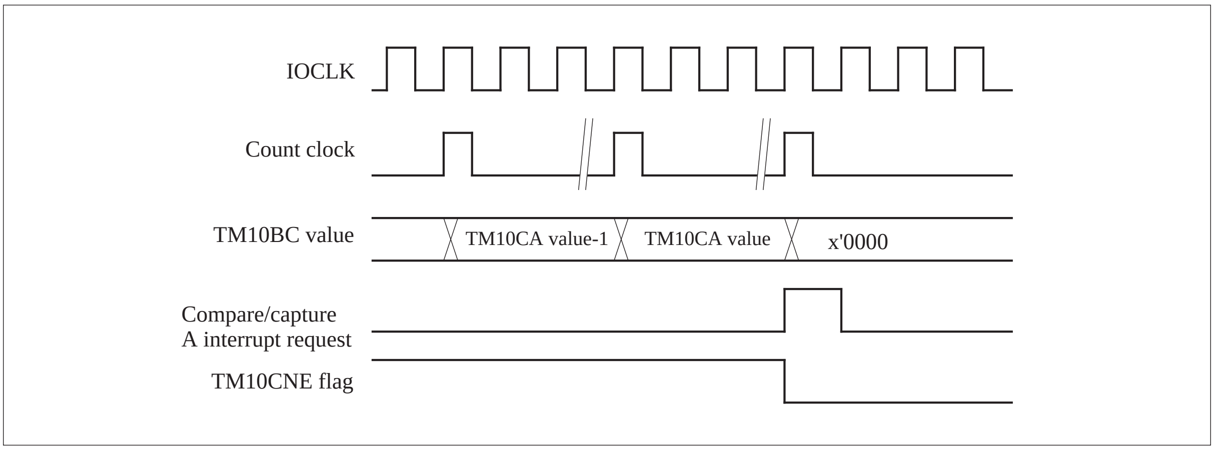


Fig. 11-6-10 One-shot Operation (When Using Prescaler)

11.6.6 Interval Timer

When using timer 10 as an interval timer, make the settings according to the procedure described below.
This interval timer generates a compare/capture A interrupt request on the cycle that is set. (Refer to Figs. 11-6-11 to 11-6-14.)
The compare/capture B register can be used as a compare register or as a capture register.

Note: For details on the settings, refer to section 11.6.1, "Compare Register Settings," or section 11.6.2, "Capture Register Settings."

■ Procedure for initiating operation

(1) Set the compare/capture A register mode.
Set the TM10MDA register as follows:

TM10AO2,1,0	Don't care
TM10ACE	0: Capture operation disabled
TM10AEG	Don't care
TM10AM1,0	00: Compare register (single-buffer) or 01: Compare register (double-buffer) If the interrupt cycle will change while the counting operation is in progress, be certain to set "double-buffer."

(2) Set the timer division ratio.
Set the division ratio in TM10CA.
The compare/capture A interrupt cycle then becomes:
(value set in TM10CA + 1) x clock source cycle

(3) Set the operating mode.
Set the TM10MD register as described below:

TM10CK2,1,0	Don't care;	Select any clock source.
TM10CAE	1:	Clears TM10BC when TM10CA matches TM10BC.
TM10ONE	0:	Disables one-shot operation.
TM10TGE	0:	Disables timer start by an external trigger.
TM10PM1,0	Don't care;	This setting is ignored.
TM10PME	0:	Selects the normal waveform.
TM10LDE	0:	Normal operation.
TM10CNE	0:	Stops counting operation.

When using 1/8IOCLK or 1/32IOCLK as the clock source, set TMPSCNE in the TMPSCNT register to "1" to enable prescaler operation before enabling the counting operation for timer 10.

(4) Initialize the timer.
Set TM10LDE in the TM10MD register to "1" in order to initialize timer 10.
TM10BC is cleared, and the pin output is reset.
In addition, if TM10CA is set as a double-buffer compare register, the value in the buffer is loaded into the compare register.
After initialization is completed, be certain to reset TM10LDE back to "0" in order to restore normal operation mode.

(5) Set the I/O port (when using pin output).
Set the I/O port to "timer output pin."
In the I/O port register, select "timer output" for the output signal and then set the output pin.

For details on the I/O port register settings, refer to chapter 15, "I/O Ports."

(6) Enable the timer counting operation.
The counting operation starts when the TM10CNE in the TM10MD register is set to "1".

Once the counting operation is enabled, a compare/capture A interrupt request is generated on a regular cycle.

If the value in the TM10CA register is changed while the counting operation is in progress, the value in the buffer is loaded into the compare register the next time that TM10BC is cleared, and the interrupt cycle is then changed. If the interrupt cycle will be changed while the counting operation is in progress, set TM10CA as a double-buffer compare register.

■ Procedure for ending operation

(1) Stop the timer counting operation.
Set TM10CNE to "0" in the TM10MD register, stopping the counting operation.

(2) Initialize the timer, if necessary.
If TM10LDE is set to "1" in the TM10MD register, TM10BC is cleared and the timer output is reset. If the TM10CA register is set as a double-buffer, the value in the compare register buffer is loaded into the compare register.
If TM10LDE is not set to "1" after the timer is stopped, the binary counter, the compare register and the pin output are maintained as they were before the timer was stopped. If TM10CNE is set to "1" again, the count resumes from the state that was in effect immediately before the timer was stopped.

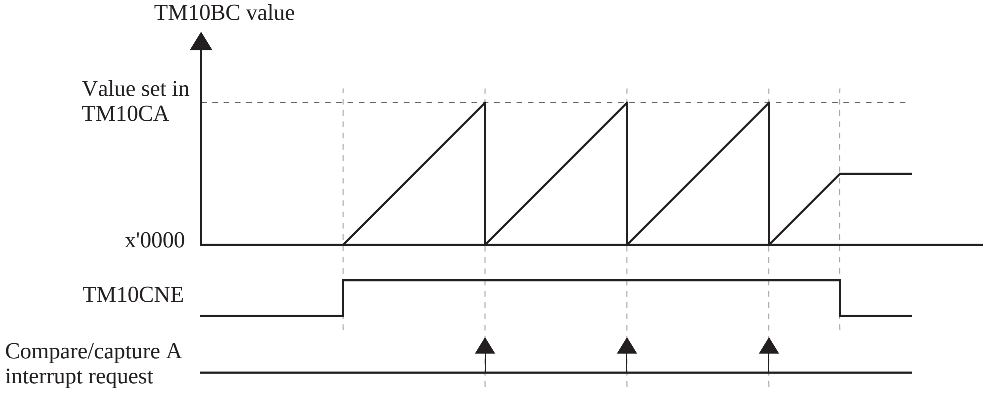


Fig. 11-6-11 Timer 10 Interval Timer Operation (1)

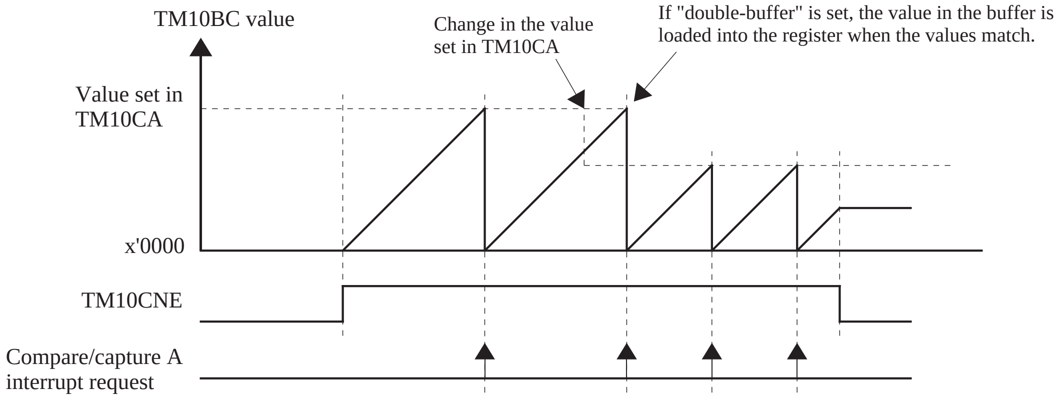

TM10BC value
Change in the value set in TM10CA
If "double-buffer" is set, the value in the buffer is loaded into the register when the values match.
Value set in TM10CA
x'0000
TM10CNE
Compare/capture A interrupt request

Fig. 11-6-12 Timer 10 Interval Timer Operation (2)

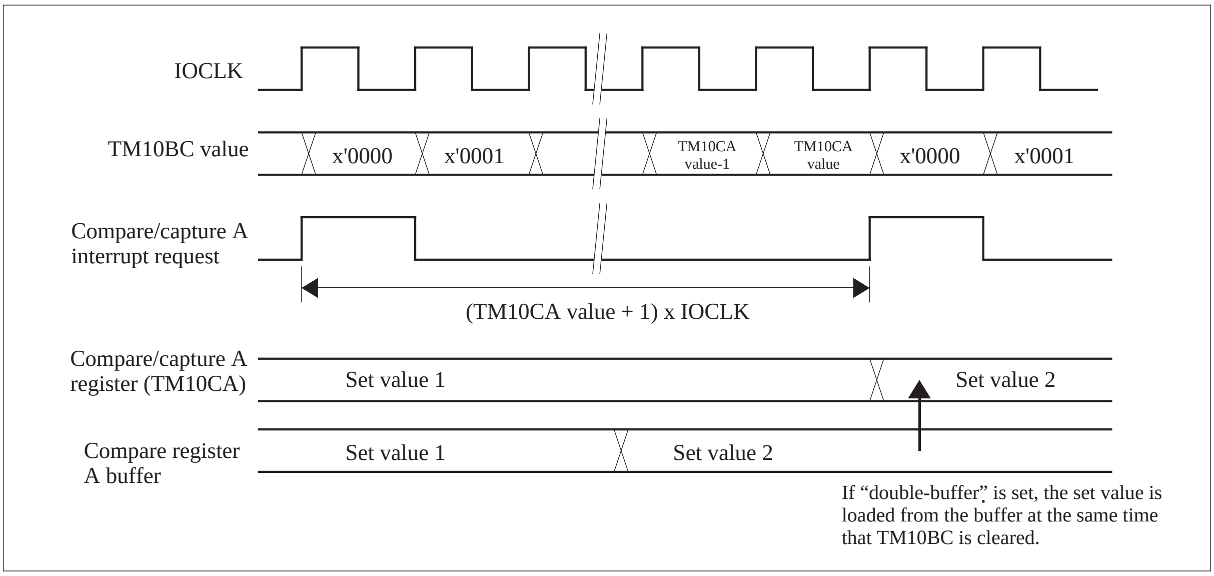

IOCLK
TM10BC value
x'0000
x'0001
TM10CA value-1
TM10CA value
x'0000
x'0001
Compare/capture A interrupt request
(TM10CA value + 1) x IOCLK
Compare/capture A register (TM10CA)
Set value 1
Set value 2
Compare register A buffer
Set value 1
Set value 2
If "double-buffer" is set, the set value is loaded from the buffer at the same time that TM10BC is cleared.

Fig. 11-6-13　Timer 10 Interval Timer Operation (When Clock Source = IOCLK)

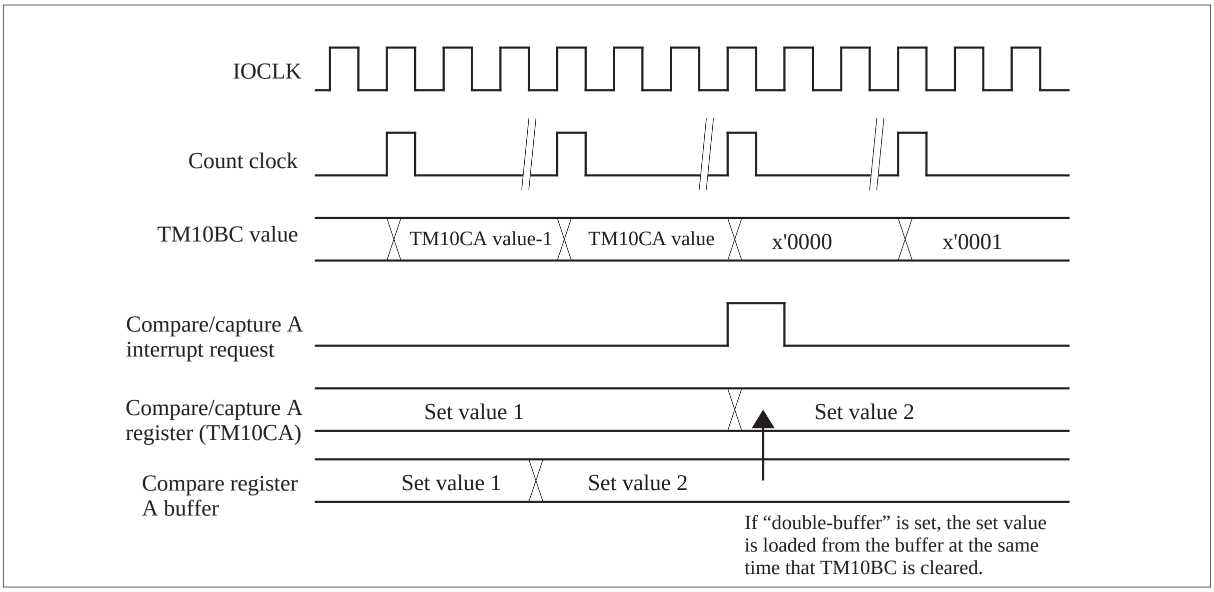

IOCLK
Count clock
TM10BC value
TM10CA value-1
TM10CA value
x'0000
x'0001
Compare/capture A interrupt request
Compare/capture A register (TM10CA)
Set value 1
Set value 2
Compare register A buffer
Set value 1
Set value 2
If "double-buffer" is set, the set value is loaded from the buffer at the same time that TM10BC is cleared.

Fig. 11-6-14　Timer 10 Interval Timer Operation (When Using Prescaler)

11.6.7 Event Counting

When using timer 10 as an event counter, make the settings according to the procedure described below.
This event counter generates a compare/capture A interrupt when it has counted the specified number of edges. (Refer to Fig. 11-6-15.)
The compare/capture B register can be used as a compare register or as a capture register.

Note: For details on the settings, refer to section 11.6.1, "Compare Register Settings."

■ Procedure for initiating operation

(1) Set the compare/capture A register mode.
Set the TM10MDA register as follows:

TM10AO2,1,0	Don't care
TM10ACE	0: Capture operation disabled
TM10AEG	Don't care
TM10AM1,0	00: Compare register (single-buffer) or 01: Compare register (double-buffer) If the interrupt cycle will change while the counting operation is in progress, be certain to set "double-buffer."

(2) Set the input edge for the TM10IOB pin.
Select either the rising edge or the falling edge at the TM10BEG of the TM10MDB register.

(3) Set the timer division ratio.
Set the division ratio in TM10CA.
A compare/capture A interrupt request is then generated when the specified edge is counted (value set in TM10CA + 1) times on the TM10IOB pin.

(4) Set the operating mode.
Set the TM10MD register as described below:

TM10CK2,1,0	111:	Sets the TM10IOB pin input as the clock source.
TM10CAE	1:	Clears TM10BC when TM10CA matches TM10BC.
TM10ONE	0:	Disables one-shot operation.
TM10TGE	0:	Disables timer start by an external trigger.
TM10PM1,0	Don't care;	This setting is ignored.
TM10PME	0:	Selects the normal waveform.
TM10LDE	0:	Normal operation.
TM10CNE	0:	Stops counting operation.

(5) Initialize the timer.
Set TM10LDE in the TM10MD register to "1" in order to initialize timer 10.
TM10BC is cleared, and the pin output is reset.
In addition, if TM10CA is set as a double-buffer compare register, the value in the buffer is loaded into the compare register.
After initialization is completed, be certain to reset TM10LDE back to "0" in order to restore normal operation mode.

(6) Set the I/O port.
Set the I/O port to "input pin."
Note:For details on the I/O port register settings, refer to chapter 15, "I/O Ports."

(7) Enable the timer counting operation.
The counting operation starts when the TM10CNE in the TM10MD register is set to "1".

Once the counting operation is enabled, TM10BC is incremented each time that the specified edge is input to the TM10IOB pin. Once (value in compare/capture A register + 1) edges are counted, TM10BC is cleared and a compare/capture A register interrupt request is generated.

If the value in the TM10CA register is changed while the counting operation is in progress, the value in the buffer is loaded into the compare register the next time that TM10BC is cleared, and the interrupt cycle is then changed. If the interrupt cycle will be changed while the counting operation is in progress, set TM10CA as a double-buffer compare register.

■ Procedure for ending operation

(1) Stop the timer counting operation.
Set TM10CNE to "0" in the TM10MD register, stopping the counting operation.

(2) Initialize the timer, if necessary.
If TM10LDE is set to "1" in the TM10MD register, TM10BC is cleared and the timer output is reset. If the TM10CA register is set as a double-buffer, the value in the compare register buffer is loaded into the compare register.
If TM10LDE is not set to "1" after the timer is stopped, the binary counter, the compare register and the pin output are maintained as they were before the timer was stopped. If TM10CNE is set to "1" again, the count resumes from the state that was in effect immediately before the timer was stopped.

[Note]
The pin input is sampled according to IOCLK. Input a signal with a pulse width of at least 6, 3, or 1.5 SYSCLK cycles when (MCLK frequency/SYSCLK frequency) = 1, 2, or 4, respectively.
Also note that event counting is not possible when IOCLK is stopped (in HALT or STOP mode).

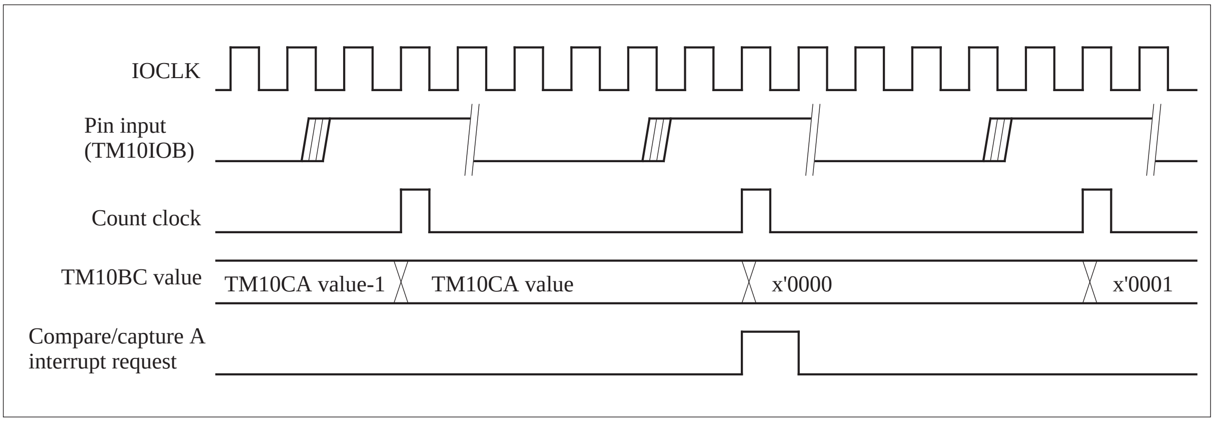


Fig. 11-6-15 Event Count Operation (When “Rising Edge” is Selected)

11.7 Description of Operation of Timers 11, 12, and 13

This section describes the operation of timers 11, 12, and 13.
Timers 11, 12, and 13 have built-in registers for setting the initial values, and down-counters. These timers can be used as interval timers and as event counters.

11.7.1 Interval Timer and Timer Output

When using timers 11, 12, or 13 as an interval timer, make the settings according to the procedure described below. These interval timers generate interrupts on the cycle that is set. (Refer to Figs. 11-7-1 to 11-7-3.)

■ Procedure for initiating operation

(1) Set the timer division ratio.
Set the division ratio in TMnBR.
The interrupt cycle is then
(value set in TMnBR + 1) x clock source cycle.

(2) Select the clock source.
Select the clock source through TMnCK[2:0] in the TMnMD register. When using 1/8IOCLK or 1/32IOCLK as the clock source, set TMPSCNE in the TMPSCNT register to "1" to enable prescaler operation before enabling the counting operation for timers 11, 12, or 13.

(3) Initialize the timer.
Set TMnLDE to "1" in the TMnMD register to initialize timer n.
The value set in TMnBR is loaded into TMnBC as the initial value, and the timer output is reset.
After initialization, be certain to set TMnLDE to "0" to return to normal operation mode.

(4) Set the I/O port (when using timer output).
Set the I/O port to "timer output pin."
In the I/O port register, select "timer output" for the output signal and then set the output pin.
For details on the I/O port register settings, refer to chapter 15, "I/O Ports."

(5) Enable the timer counting operation.
The counting operation starts when the TMnCNE in the TMnMD register is set to "1".

Once the counting operation is enabled, an underflow interrupt request is generated on a regular cycle. In addition, with each interrupt the pin output is inverted and the value in TMnBR is loaded into TMnBC.
If the value in the TMnBR register is changed while the counting operation is in progress, this changed value is loaded as the initial value the next time that an underflow is generated, and the interrupt cycle is then changed.

■ Procedure for ending operation

(1) Stop the timer counting operation.
Set TMnCNE to "0" in the TMnMD register, stopping the counting operation.

(2) Initialize the timer, if necessary.
If TMnLDE is set to "1" in the TMnMD register, the value that is set in TMnBR is loaded into TMnBC as the initial value, and the timer output is reset. If TMnLDE is not set to "1" after the timer is stopped, the binary counter and the pin output are maintained as they were before the timer was stopped. If TMnCNE is set to "1" again, the count resumes from the state that was in effect immediately before the timer was stopped.

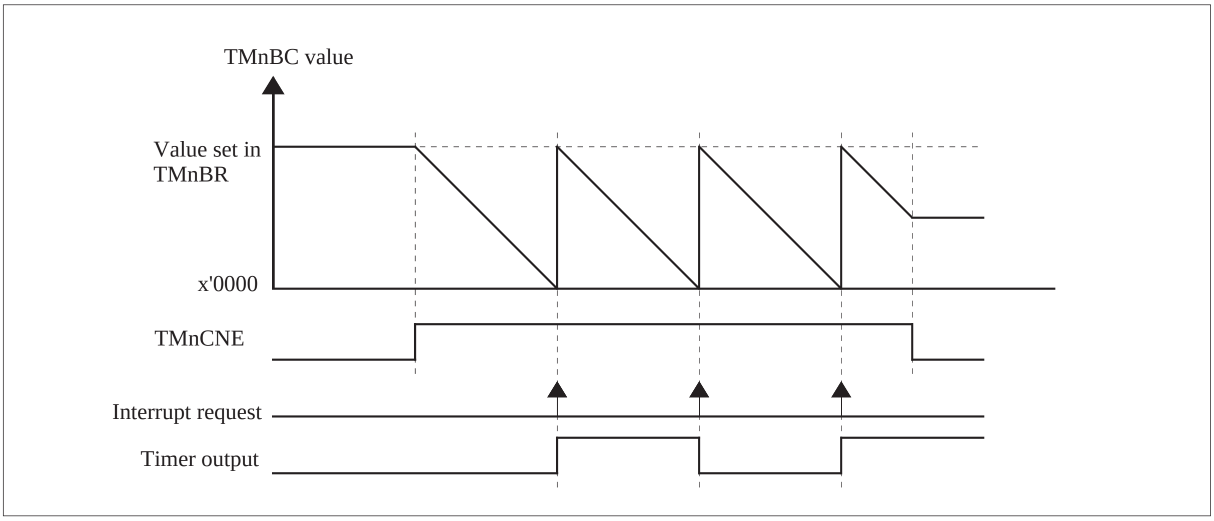


Fig. 11-7-1 Interval Timer Operation

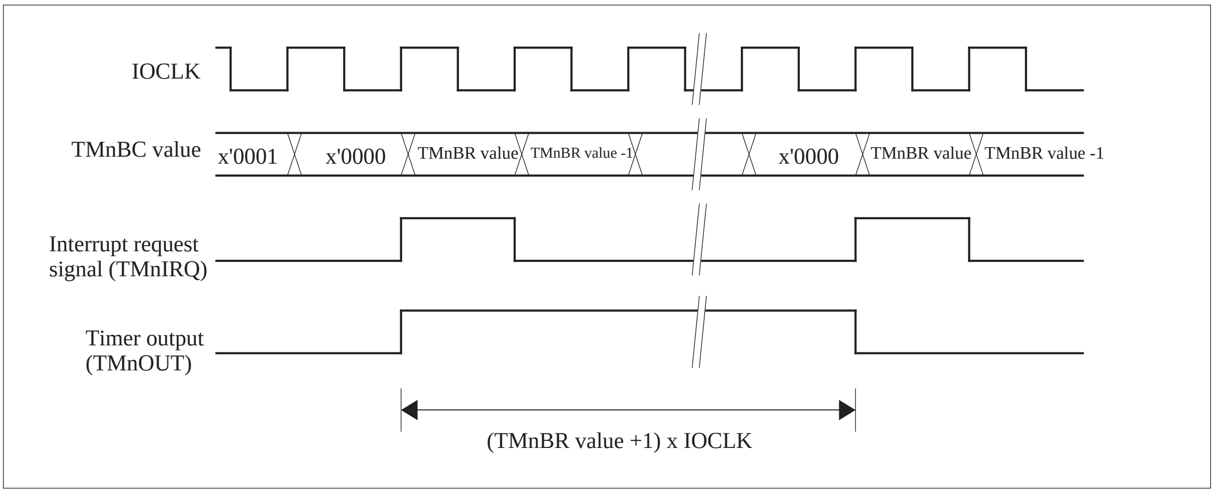


Fig. 11-7-2 Interval Timer Operation (When Clock Source = IOCLK)

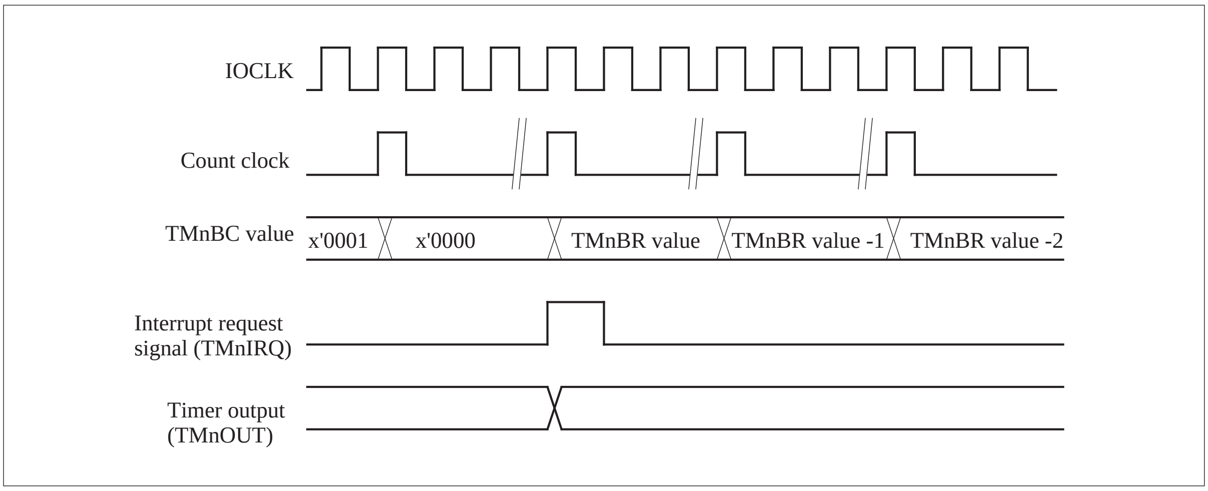


Fig. 11-7-3 Interval Timer Operation (When Using the Prescaler)

11.7.2 Event Counting

When using timer 11, 12, or 13 as an event counter, make the settings according to the procedure described below.

■ Procedure for initiating operation

(1) Set the timer division ratio.
Set the division ratio in TMnBR.
An interrupt request is then generated when the rising edge is counted (value set in TMnBR + 1) times in the pin input.

(2) Select the clock source.
Select the clock source through TMnCK[2:0] in the TMnMD register to "TMnIO pin input."

(3) Initialize the timer.
Set TMnLDE to "1" in the TMnMD register to initialize timer n.
The value set in TMnBR is loaded into TMnBC as the initial value.
After initialization, be certain to set TMnLDE to "0" to return to normal operation mode.

(4) Set the I/O port.
Set the I/O port to "input pin."
For details on the I/O port register settings, refer to chapter 15, "I/O Ports."

(5) Enable the timer counting operation.
The counting operation is enabled when the TMnCNE in the TMnMD register is set to "1".

Once the counting operation is enabled, the counter counts rising edges on the pin input. When an underflow occurs in the binary counter, an interrupt is generated and the value set in TMnBR is loaded into TMnBC. (Refer to Fig. 11-7-4.)
If the value in the TMnBR register is changed while the counting operation is in progress, this changed value is loaded as the initial value the next time that an underflow is generated.

■ Procedure for ending operation

(1) Stop the timer counting operation.
Set TMnCNE to "0" in the TMnMD register, stopping the counting operation.

(2) Initialize the timer, if necessary.
If TMnLDE is set to "1" in the TMnMD register, the value that is set in TMnBR is loaded into TMnBC as the initial value.
If TMnLDE is not set to "1" after the timer is stopped, the binary counter is maintained as it was before the timer was stopped.
If TMnCNE is set to "1" again, the count resumes from the state that was in effect immediately before the timer was stopped.

[Note]
The pin input is sampled according to IOCLK. Input a signal with a pulse width of at least 6, 3, or 1.5 SYSCLK cycles when (MCLK frequency/SYSCLK frequency) = 1, 2, or 4, respectively.
Also note that event counting is not possible when IOCLK is stopped (in HALT or STOP mode).

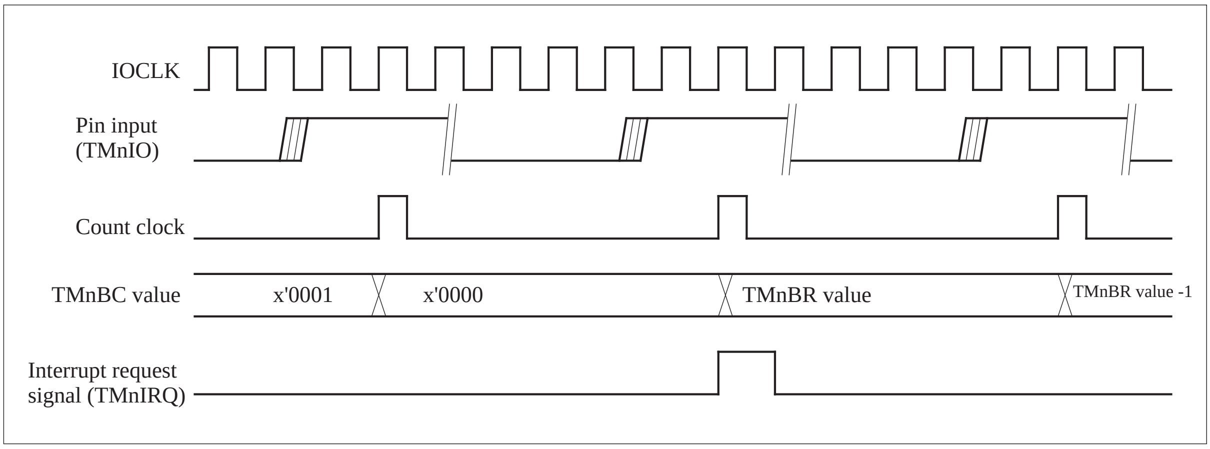


Fig. 11-7-4   Event Count Operation

12. Watchdog Timer

12.1 Overview

This microcontroller has a 25-bit binary counter built in that can be used as a 16- to 25-bit watchdog timer.
A watchdog timer overflow generates a nonmaskable interrupt, enabling the watchdog timer overflow to be identified.
The watchdog timer is also used as an oscillation stabilization wait timer.

12.2 Features

- The number of bits in the binary counter is selectable.
  When the CKSEL pin input is "H" (oscillating frequency: 8 MHz to 18 MHz):
  16, 18, 20, 22, or 24 bits can be selected.
  When the CKSEL pin input is "L" (oscillating frequency: 8 MHz to 20 MHz):
  17, 19, 21, 23, or 25 bits can be selected.
  Overflow cycle: 4.369 ms to 1118.481 ms
  (when the CKSEL pin input is "H" and the oscillating frequency is 15 MHz)
- A non-maskable interrupt is generated when a watchdog timer overflow occurs.
- Watchdog timer overflow output
  A flag can be set to "1" when a watchdog timer overflow occurs.
  The watchdog timer overflow output can be selected as either pulse output or level output.
- Oscillation stabilization wait time (when the CKSEL pin input is "H" and the oscillating frequency is 15 MHz)
  When reset is released: 17.476 ms
  When recovering from STOP mode: 4.369 ms to 1118.481 ms <Recommended value is 14 ms or longer.>
- The chip can self-reset internally by writing the RSTCTR register.

12.3 Block Diagram

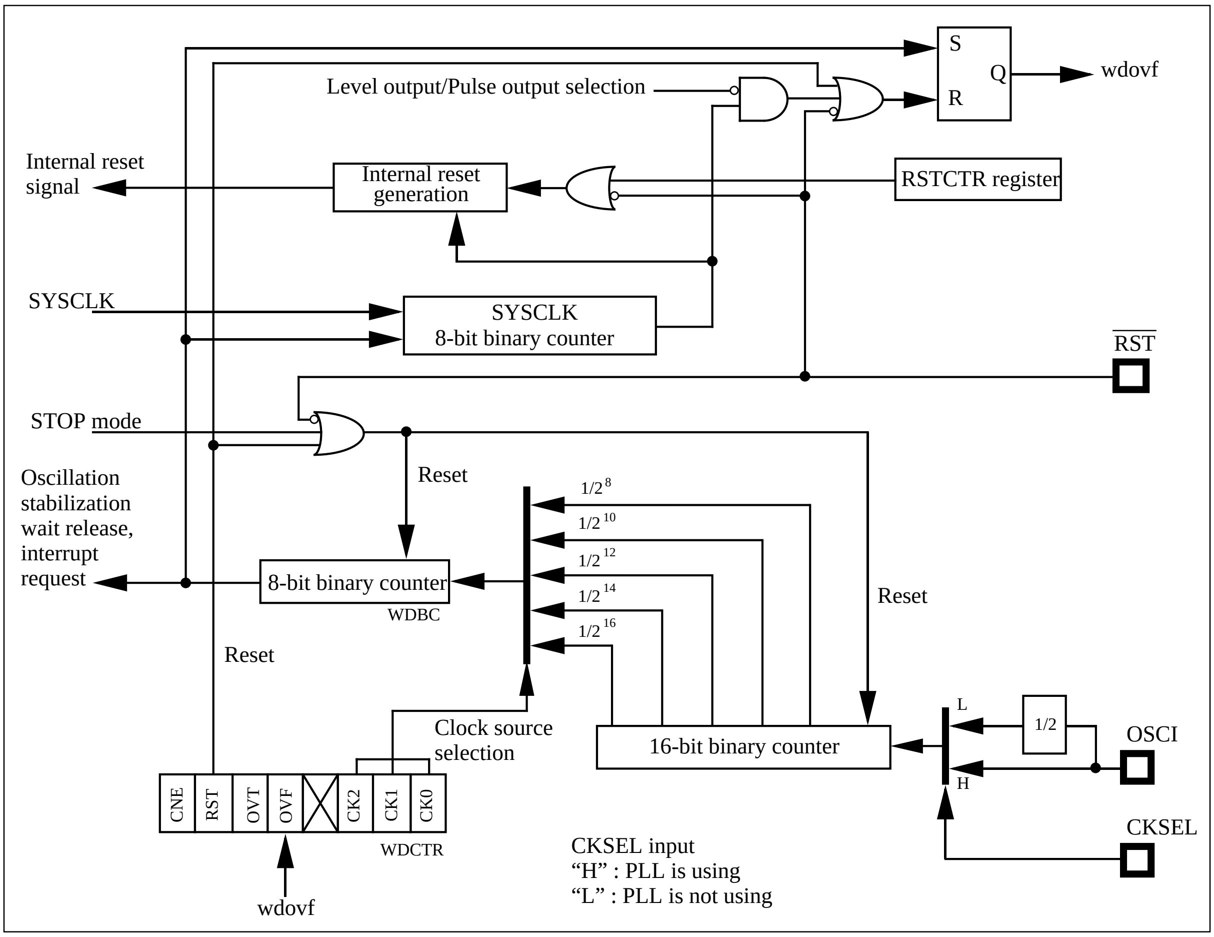


Fig. 12-3-1 Block Diagram

12.4 Description of Registers

Table 12-4-1 lists the watchdog timer registers.

Table 12-4-1 List of Watchdog Timer Registers

Address	Name	Symbol	Number of bits	Initial value	Access size
x'34004000	Watchdog binary counter	WDBC	8	x'00	8
x'34004008	Watchdog timer control register	WDCTR	8	x'01	8
x'34004004	Reset control register	RSTCTR	8	x'00	8

Watchdog binary counter

Register symbol: WDBC
Address: x'34004000
Purpose: Reading this counter returns the counter value of the high-order eight bits of the watchdog timer.

Bit No.	7	6	5	4	3	2	1	0
Bit name	WD BC7	WD BC6	WD BC5	WD BC4	WD BC3	WD BC2	WD BC1	WD BC0
Reset	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R

Bit No.	Bit name	Description
7 to 0	WDBC7 to 0	Counter value of the high-order eight bits of the watchdog timer The value that is read is not guaranteed if the value is changed while it is being read.

Watchdog timer control register

Register symbol: WDCTR
Address: x'34004008
Purpose: This register sets the watchdog timer operation control conditions.

Bit No.	7	6	5	4	3	2	1	0
Bit name	WD CNE	WD RST	WD OVT	WD OVF	-	WD CK2	WD CK1	WD CK0
Reset	0	0	0	0	0	0	0	1
Access	R/W	R/W	R/W	R	R	R/W	R/W	R/W

Bit No.	Bit name	Description
0	WDCK0	Clock source selection (LSB)
1	WDCK1	Clock source selection
2	WDCK2	Clock source selection (MSB)

These bits select the clock source for the high-order 8 bits of the counter. When the reset state is released, the clock source corresponding to "001" below is selected.

	When CKSEL is "H"	When CKSEL is "L"
000:	$1/2^{8}$ of the OSCI input	$1/2^{9}$
001:	$1/2^{10}$ of the OSCI input	$1/2^{11}$
010:	$1/2^{12}$ of the OSCI input	$1/2^{13}$
011:	$1/2^{14}$ of the OSCI input	$1/2^{15}$
100:	$1/2^{16}$ of the OSCI input	$1/2^{17}$
101:	Setting prohibited	
110:	Setting prohibited	
111:	Setting prohibited	

Overflow cycle = $2^{(n + \text{WDCK x } 2)}/(f \text{ x } 10^{3})$ [ms]
Where, n = 16 (CKSEL pin is “H”) or n = 17 (CKSEL pin is “L”);
WDCK = WDCK[2:0]; f: Oscillation input frequency [unit: MHz]

Example

Selection	Overflow cycle When CKSEL is "H" and oscillating frequency is 15 MHz
000	4.369 ms
001	17.476 ms
010	69.905 ms
011	279.620 ms
100	1118.481 ms

3 — When this bit is read, a "0" is returned.

4 WDOVF The value of the watchdog timer overflow output.

5 WDOVT Watchdog timer overflow output selection
0: Pulse output
1: Level output

6 WDRST Binary counter reset, watchdog timer overflow output (WDOVF flag) reset
0: No reset
1: Reset
When a "1" is written to this bit, the reset pulse is generated for the width of one clock pulse, and then this bit returns to "0". "0" is returned whenever this bit is read.

7 WDCNE Watchdog timer count operation control flag
0: Count operation stopped
(oscillation stabilization wait operation is possible)
1: Count operation enabled

[Notes]

1. When resetting the value of watchdog overflow by writing the WDRST flag, do not simultaneously overwrite the WDOVT flag.
   If this flag is overwritten, the value of watchdog overflow reset is not guaranteed.
2. When changing the values of WDCK2 to 0, first stop the watchdog timer and reset the counter.

Reset control register

Register symbol: RSTCTR
Address: x'34004004
Purpose: This flag causes the program to generate an internal reset.

Bit No.	7	6	5	4	3	2	1	0
Bit name	-	-	-	-	-	-	-	CHIP RST
Reset	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R/W

Bit No. Bit name Description

0 CHIPRST This flag is used to generate a self-reset (an internal reset of the chip due to an internal cause). A self-reset is generated when this flag is overwritten from "0" to "1".
A self-reset is not generated if this flag is already set to "1" when it is written with a "1".
The value stored in this flag is retained even after the reset.
The CHIPRST flag is cleared either by an external reset signal ($\overline{\text{RST}}$) or by writing a "0" to this flag through the software.

12.5 Description of Operation

Oscillation stabilization wait operation

The watchdog timer operates as an oscillation stabilization wait timer after the reset state is released or when the microcontroller recovers from STOP mode (Fig. 12-5-1).

The watchdog timer operates in this capacity even if the WDCNE flag is "0".

When recovering from STOP mode, the watchdog timer operates as a counter of the number of bits specified by WDCK2 to 0 (Fig. 12-5-2). The oscillation stabilization wait time can be selected from among times that are calculated as follows:

Overflow cycle = $2^{(n + \mathrm{WDCK} \times 2)}$/(f x 10^3) [ms]

Where, n = 16 (CKSEL pin is “H”) or n = 17 (CKSEL pin is “L”); WDCK = WDCK[2:0]; f: Oscillation input frequency [unit: MHz]

An oscillation stabilization wait time of at least 14 ms is recommended.

If the WDCNE flag is "1", a non-maskable interrupt is not generated even when recovering from STOP mode.

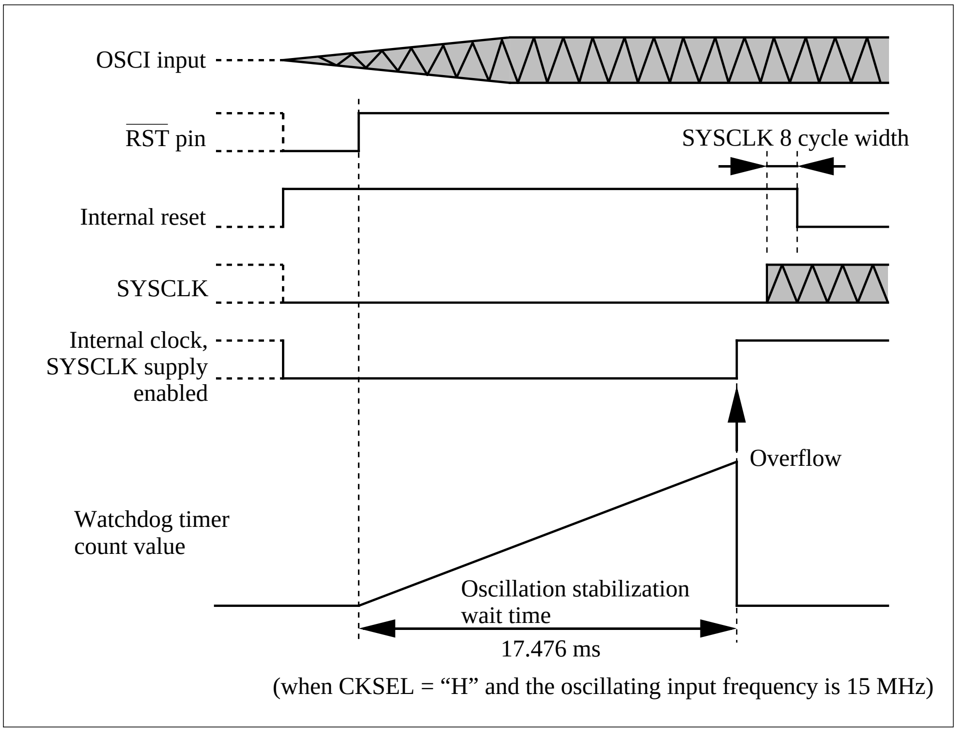

OSCI input
$\overline{\text{RST}}$ pin
SYSCLK 8 cycle width
Internal reset
SYSCLK
Internal clock, SYSCLK supply enabled
Overflow
Watchdog timer count value
Oscillation stabilization wait time
17.476 ms
(when CKSEL = “H” and the oscillating input frequency is 15 MHz)

Fig. 12-5-1 Operation Diagram 1: When Reset Is Released

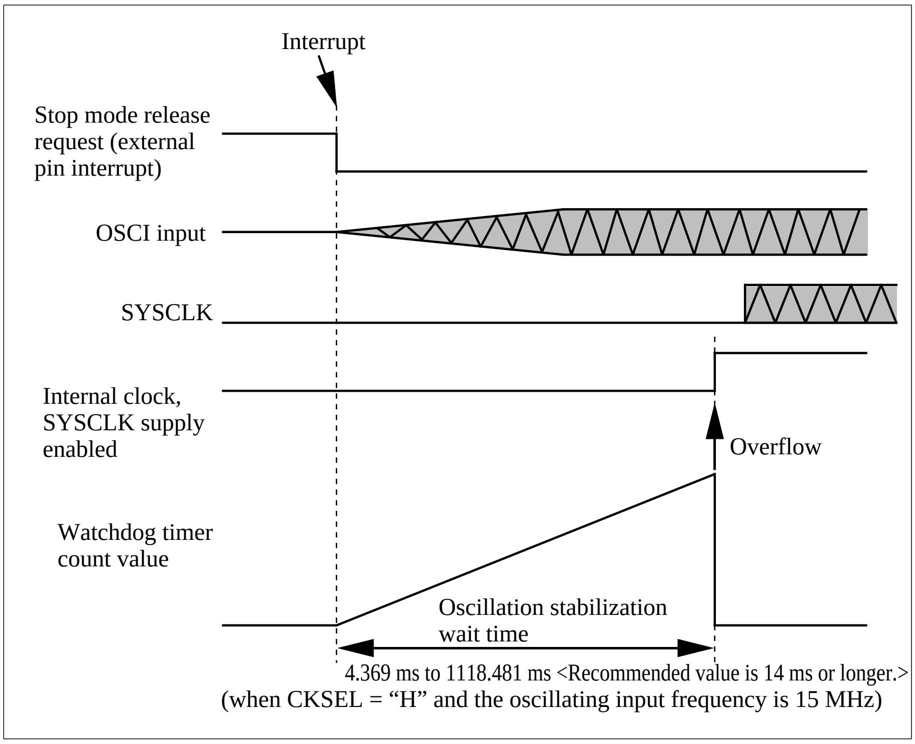

Interrupt
Stop mode release request (external pin interrupt)
OSCI input
SYSCLK
Internal clock, SYSCLK supply enabled
Overflow
Watchdog timer count value
Oscillation stabilization wait time
4.369 ms to 1118.481 ms <Recommended value is 14 ms or longer.>
(when CKSEL = “H” and the oscillating input frequency is 15 MHz)

Fig. 12-5-2 Operation Diagram 2: When Recovering from STOP Mode

Watchdog operation

If the WDCNE flag is set to "1" and the watchdog operation is enabled, a non-maskable interrupt is generated if a watchdog timer overflow occurs.

When an overflow occurs, the watchdog timer overflow output is output to the WDOVF flag. Pulse output or level output can be selected through the WDOVT flag. When level output is selected, the watchdog timer overflow output (WDOVF flag) is cleared by writing a "1" to the WDRST flag or by reset ($\overline{\text{RST}}$) pin "L" level input.

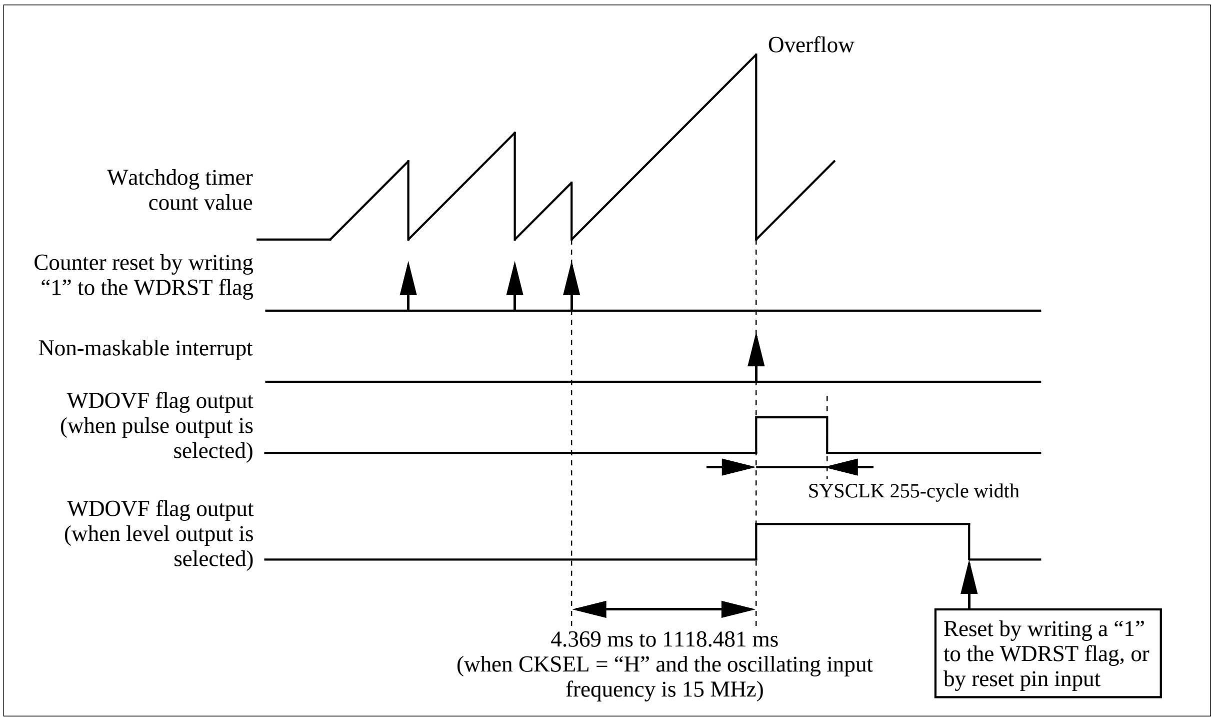

Overflow
Watchdog timer count value
Counter reset by writing "1" to the WDRST flag
Non-maskable interrupt
WDOVF flag output (when pulse output is selected)
SYSCLK 255-cycle width
WDOVF flag output (when level output is selected)
4.369 ms to 1118.481 ms (when CKSEL = "H" and the oscillating input frequency is 15 MHz)
Reset by writing a "1" to the WDRST flag, or by reset pin input

Fig. 12-5-3 Operation Diagram 3: Watchdog Operation

Before setting the WDCNE flag to "1", write a "1" to the WDRST flag to reset the counter.
When switching to HALT or SLEEP mode, set the WDCNE flag to "0" to turn off the watchdog timer.

Self-reset operation

The chip resets internally when a "1" is written to the CHIPRST bit in the RSTCTR register. The oscillation stabilization wait operation is not performed.

The reset generated by writing the CHIPRST flag is an internal reset signal within the chip and does not manifest itself on the external reset pin ($\overline{\text{RST}}$ pin).

13. Serial Interface

13.1 Overview

This microcontroller has three types of internal serial interfaces. One is a general-purpose serial interface for which clock synchronous mode, UART mode, or I2C mode can be specified; this interface supports one channel. The second interface is a clock synchronous serial interface that supports two channels. The third interface is UART serial interface that supports one channel.

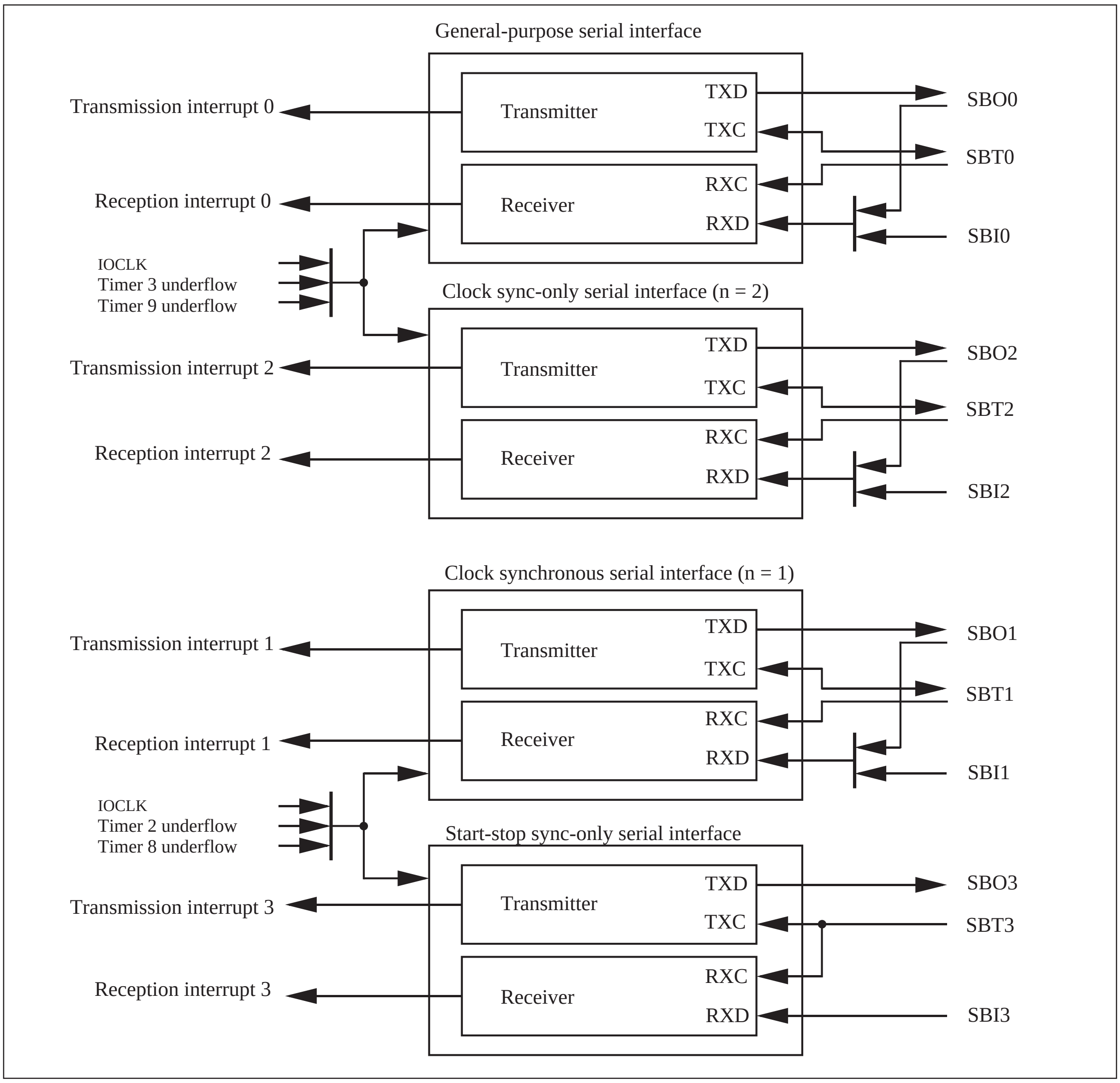


Fig. 13-1-1 Structure Diagram

13.2 General-purpose serial interface

13.2.1 Features

Serial interface 0 is a general-purpose serial interface for which clock sync mode, UART mode, or I2C mode can be specified. The features of each mode are described below.

<Clock synchronous mode>

- Parity None, 0 fixed, 1 fixed, even, odd
- Character length 7 bits, 8 bits
- Transmission and reception bit sequence
  LSB or MSB selectable
- Clock source 1/2, 1/8, or 1/32 of IOCLK
  1/8 of timer 3 or timer 9 underflow, 1/2 of timer 9 underflow
  External clock
- Maximum bit rate
  7.5 Mbit/s (when IOCLK is 15 MHz)
- Error detection during reception
  Parity errors, overrun errors
- Buffers Independent buffers for transmission and reception:
  Reception and transmission buffers are both double buffers
- Interrupts Transmission interrupts:
  "Transmission end" or "transmission buffer empty" selectable
  Reception interrupts:
  "Reception end" or "reception end with error" selectable

<UART mode>

- Parity None, 0 fixed, 1 fixed, even, odd
- Character length 7 bits, 8 bits
- Transmission and reception bit sequence
  LSB or MSB selectable
- Clock source 1/8 or 1/32 of IOCLK
  1/8 of timer 3 or timer 9 underflow
  1/8 of external clock
- Maximum bit rate
  19.2 kbit/s (when IOCLK is 15 MHz)
- Error detection during reception
  Parity errors, overrun errors, framing errors
- Buffers Independent buffers for transmission and reception
  Reception and transmission buffers are both double buffers
- Interrupts Transmission interrupts:
  "Transmission end" or "transmission buffer empty" selectable
  Reception interrupts:
  "Reception end" or "reception end with error" selectable

<I2C mode>

- Master transmission, master reception possible (No start sequence conflict detection function)

13.2.2 Block Diagram of General-Purpose Serial Interface

Fig 13-2-1 shows the block diagram for the general-purpose serial interface section.

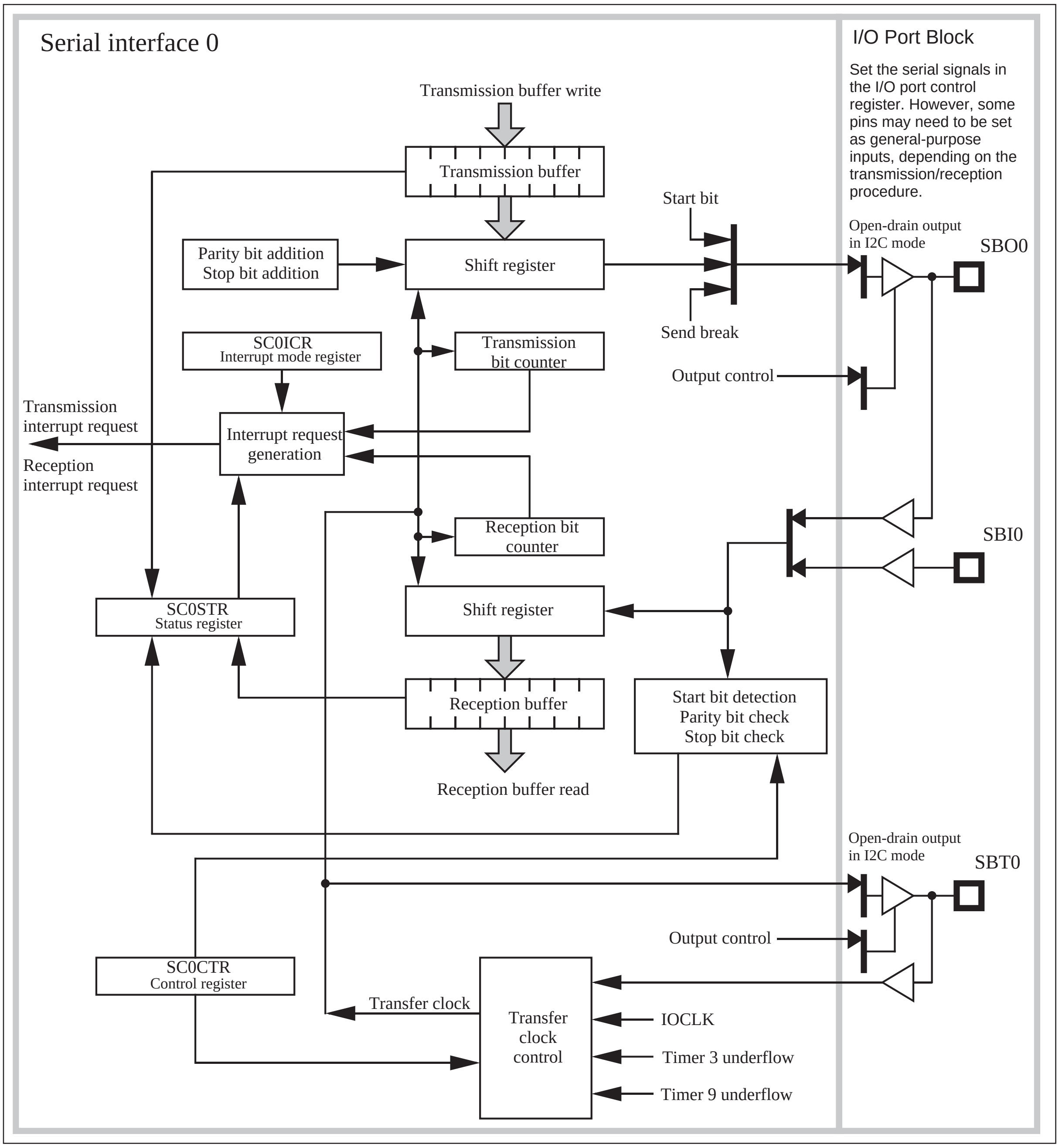


Fig. 13-2-1 Block Diagram

13.2.3 Description of Registers for the General-Purpose Serial Interface

The general-purpose serial interface includes the registers listed in Table 13-2-1. These registers are used for settings such as clock source selection, parity bit selection, and protocol selection.

Table 13-2-1 List of General-Purpose Serial Interface Registers

Address	Name	Symbol	Number of bits	Initial value	Access size
x'34000800	Serial 0 control register	SC0CTR	16	x'0000	8, 16
x'34000804	Serial 0 interrupt mode register	SC0ICR	8	x'00	8
x'34000808	Serial 0 transmission buffer	SC0TXB	8	x'00	8
x'34000809	Serial 0 reception buffer	SC0RXB	8	x'00	8
x'3400080C	Serial 0 status register	SC0STR	16	x'0000	8, 16

Serial 0 control register

Register symbol: SC0CTR
Address: x'34000800
Purpose: This register sets the serial interface 0 operation control conditions.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	SC0 TXE	SC0 RXE	SC0 BKE	SC0 IIC	SC0 MD1	SC0 MD0	SC0 OD	SC0 TOE	SC0 CLN	SC0 PB2	SC0 PB1	SC0 PB0	SC0 STB	SC0 CK2	SC0 CK1	SC0 CK0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit No.	Bit name	Description
0	SC0CK0	Clock source selection (LSB)
1	SC0CK1	Clock source selection
2	SC0CK2	Clock source selection (MSB) 000: 1/2 IOCLK (valid only in clock synchronous mode) 001: 1/8 IOCLK 010: 1/32 IOCLK 011: 1/2 timer 9 underflow (valid only in clock synchronous mode) 100: 1/8 timer 3 underflow 101: 1/8 timer 9 underflow 110: 1/8 external clock (valid only in UART mode) 111: External clock (valid only in clock synchronous mode)
3	SC0STB	Stop bit selection (valid only in UART mode) 0: 1 bit 1: 2 bits
4	SC0PB0	Parity bit selection (LSB)
5	SC0PB1	Parity bit selection

6	SC0PB2	Parity bit selection (MSB) 000: None 001, 010, 011: Setting prohibited 100: 0 fixed 101: 1 fixed 110: Even (even number of ones) 111: Odd (odd number of ones)
7	SC0CLN	Character length selection 0: 7 bits 1: 8 bits
8	SC0TOE	SBT0 pin output control 0: When the internal clock is selected, the SBT0 pin is an output only while transmission is in progress (the SBT0 pin is an input when in standby mode or when an external clock is selected) 1: When the internal clock is selected, the SBT0 pin is always an output (the SBT0 pin is an input when an external clock is selected)
9	SC0OD	Transmission and reception bit sequence selection 0: From LSB 1: From MSB
10	SC0MD0	Protocol selection (LSB)
11	SC0MD1	Protocol selection (MSB) 00: UART mode 01: Clock synchronous mode (1) (the SBO0 pin is used as a data output, and the SBI0 pin is used as a data input) 10: I2C mode 11: Clock synchronous mode (2) (the SBO0 pin is used as a data input and output, and input on the SBI0 pin is ignored)
12	SC0IIC	I2C mode selection 0: The stop sequence is output when this bit is changed from "1" to "0" 1: The start sequence is output when this bit is changed from "0" to "1"
13	SC0BKE	Break transmission 0: Do not send break 1: Send break (The SBO0 pin is fixed to output "0".)
14	SC0RXE	Reception operation enable 0: Disabled 1: Enabled
15	SC0TXE	Transmission operation enable 0: Disabled 1: Enabled

Serial 0 interrupt mode register

Register symbol: SC0ICR
Address: x'34000804
Purpose: This register selects the sources for transmission interrupts and reception interrupts for serial interface 0.

Bit No.	7	6	5	4	3	2	1	0
Bit name	SC0 DMD	-	-	SC0 TI	-	SC0 RES	-	SC0 RI
Reset	0	0	0	0	0	0	0	0
Access	R/W	R	R/W	R/W	R	R/W	R	R/W

Bit No.	Bit name	Description
0	SC0RI	Reception interrupt factor selection 0: Reception end 1: Reception end with error
1	—	"0" is returned when this bit is read.
2	SC0RES	Reception error interrupt factor selection 0: Interrupt request when an overrun, parity, or framing error occurs 1: Interrupt request when a parity error occurs
3	—	"0" is returned when this bit is read.
4	SC0TI	Transmission interrupt factor selection 0: Transmission end 1: Transmission buffer empty
5	Reserved	Setting "1" is prohibited.
6	—	"0" is returned when this bit is read.
7	SC0DMD	Data output retained during external clock transmission (valid only in clock synchronous mode) 0: Set data pin "H" at end of transmission 1: Maintain data pin at end of transmission

Serial 0 transmission buffer

Register symbol SC0TXB
Address: x'34000808
Purpose: This register writes the transmission data of serial interface 0.

Bit No.	7	6	5	4	3	2	1	0
Bit name	SC0 TXB7	SC0 TXB6	SC0 TXB5	SC0 TXB4	SC0 TXB3	SC0 TXB2	SC0 TXB1	SC0 TXB0
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Data is transmitted by writing it to this buffer.

Serial 0 reception buffer

Register symbol: SC0RXB
Address: x'34000809
Purpose: This register reads in the reception data of serial interface 0.

Bit No.	7	6	5	4	3	2	1	0
Bit name	SC0 RXB7	SC0 RXB6	SC0 RXB5	SC0 RXB4	SC0 RXB3	SC0 RXB2	SC0 RXB1	SC0 RXB0
Reset	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R

Reception data is gotten by reading this buffer at the end of reception.
In the case of a 7-bit transfer, the MSB (bit 7) is "0".

Serial 0 status register

Register symbol: SC0STR
Address: x'3400080C
Purpose: This register indicates the status of serial interface 0.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	-	-	-	-	-	-	SC0 SPF	SC0 STF	SC0 TXF	SC0 RXF	SC0 TBF	SC0 RBF	-	SC0 FEF	SC0 PEF	SC0 OEF
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit No. Bit name Description

0 SC0OEF Overrun error indication
0: No error 1: Overrun error occurred

1 SC0PEF Parity error indication
0: No error 1: Parity error occurred

2 SC0FEF Framing error indication
0: No error 1: Framing error occurred

3 — "0" is returned when this bit is read.

4 SC0RBF Reception buffer status indication
0: Reception buffer empty
1: Data exists in the reception buffer

5 SC0TBF Transmission buffer status indication
0: Transmission buffer empty
1: Data exists in the transmission buffer

6 SC0RXF Reception status indication
0: Waiting for reception
1: Reception in progress

7 SC0TXF Transmission status indication
0: Ready for transmission
1: Transmission in progress

8 SC0STF I2C start sequence detection
(Cleared by reading SC0RXB or by writing SC0TXB)
0: Not detected
1: Detected

9 SC0SPF I2C stop sequence detection
(Cleared by reading SC0RXB or by writing SC0TXB)
0: Not detected
1: Detected

15 to 10 — "0" is returned when these bits are read.

13.2.4 Description of Operation

<Clock synchronous mode>

■ Clock synchronous mode connection

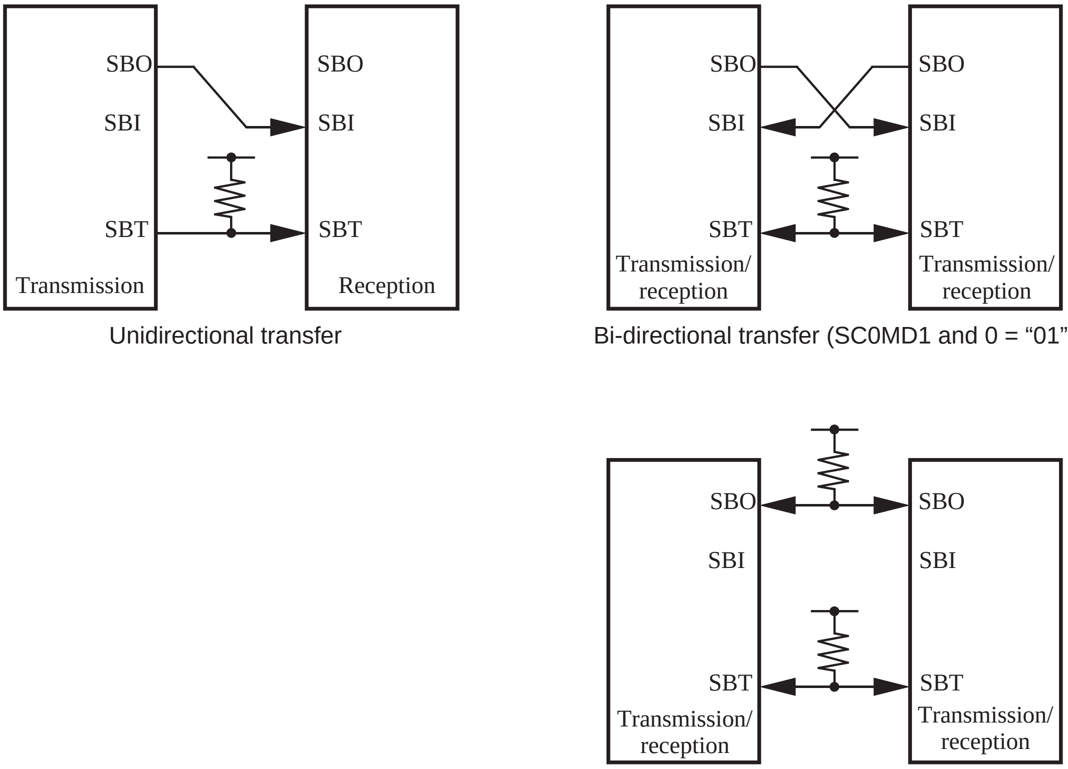
Two different connection methods are possible, one for unidirectional transfer, and the other for bi-directional transfer.

When SBT pin is an output only during transmission (SC0TOE = "0"), it is necessary to pull up SBT pin. In addition, when using SBO pin as a data input/output (SC0MD1 and 0 = "11"), it is necessary to pull up SBO pin. Connect a pull-up resistor externally.

When using SBO pin as a data output and SBI pin as a data input (SC0MD1 and 0 = "01"), the SBO pin is always an output and the SBI pin is always an input.

When using SBO pin as a data input/output (SC0MD1 and 0 = "11"), the SBO pin is an output only during transmission, and is normally an input.

When SC0TOE is "0", the SBT pin is an output only during transmission with the internal clock, and is normally an input. Furthermore, when SC0TOE is "1", the SBT pin is always an output when the internal clock is selected.



Unidirectional transfer

Bi-directional transfer (SC0MD1 and 0 = “01”)

Bi-directional transfer (SC0MD1 and 0 = “11”)

Fig. 13-2-2 Connections

■ Clock synchronous mode timing

<Transmission>

- One-byte transfer with 8-bit data length and parity on

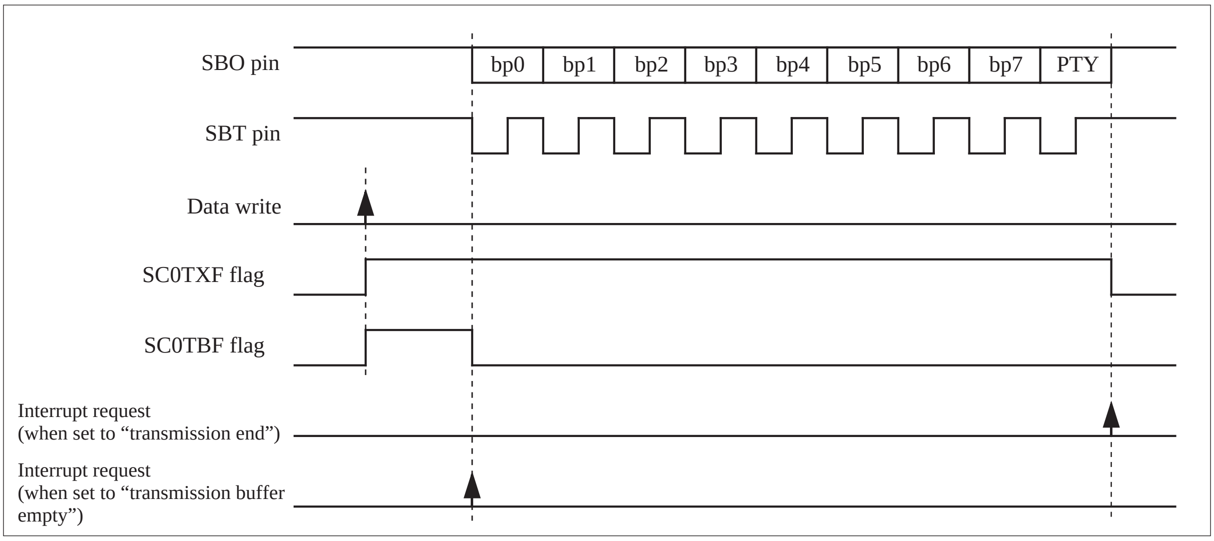


Fig. 13-2-3 Timing Chart (1)

- Two-byte transfer with 8-bit data length and parity off

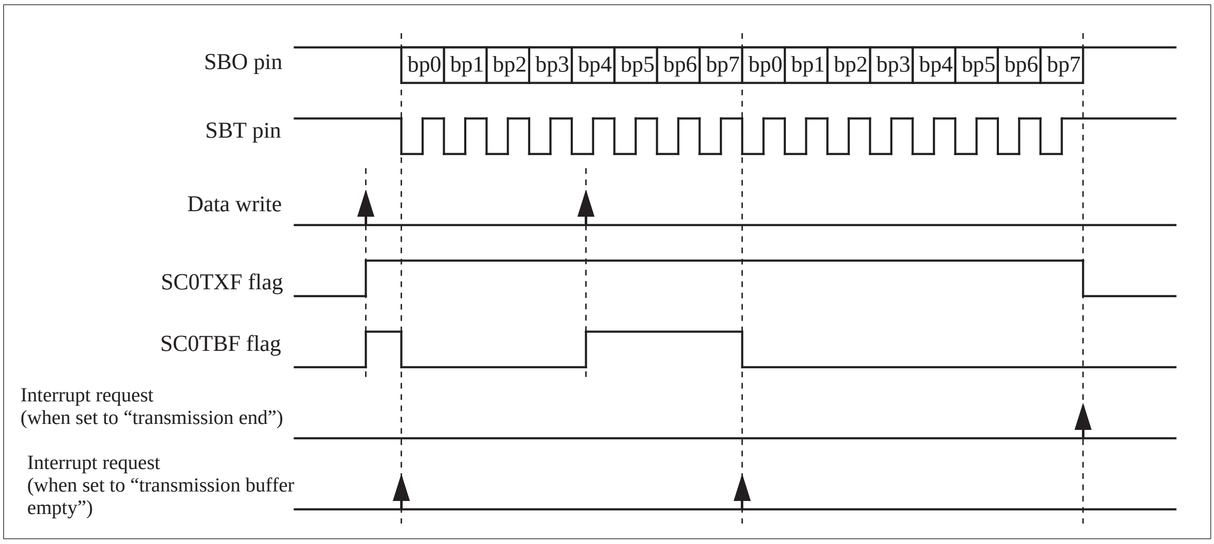


Fig. 13-2-4 Timing Chart (2)

When transmission is enabled, transmission starts when data is written to SC0TXB.
Continuous transmission is possible by writing data to SC0TXB again while transmission is in progress.
During a 7-bit transfer, the MSB (bit 7) is ignored.

The SC0TXF flag is set to "1" when data is written to SC0TXB, and is set to "0" at the end of transmission.
The SC0TBF flag is set to "1" when data is written to SC0TXB, and is set to "0" at the start of transmission.

<Reception>

- One-byte transfer with 8-bit data length and parity on

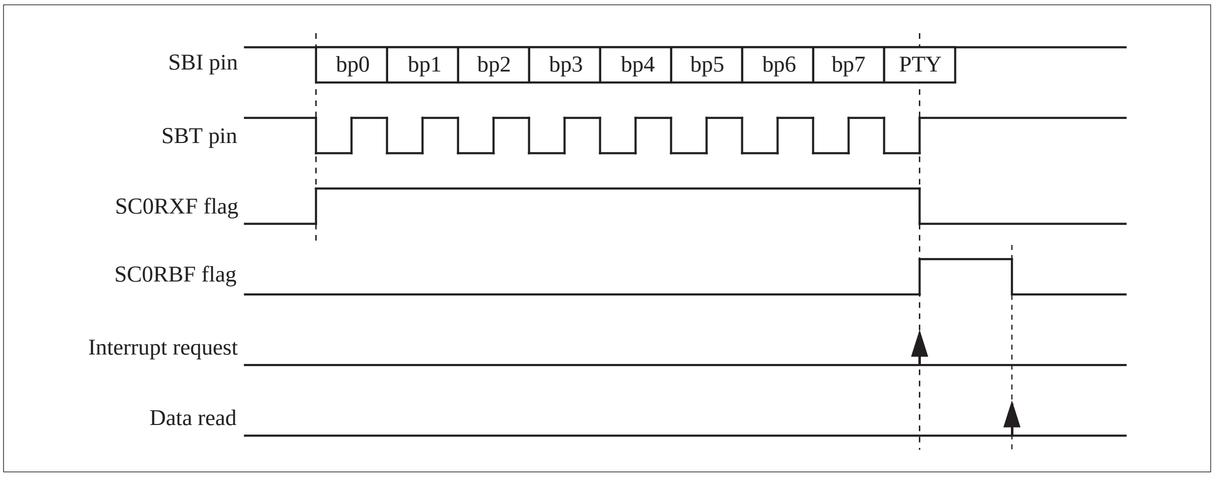


Fig. 13-2-5 Timing Chart (3)

- Two-byte transfer with 8-bit data length and parity off

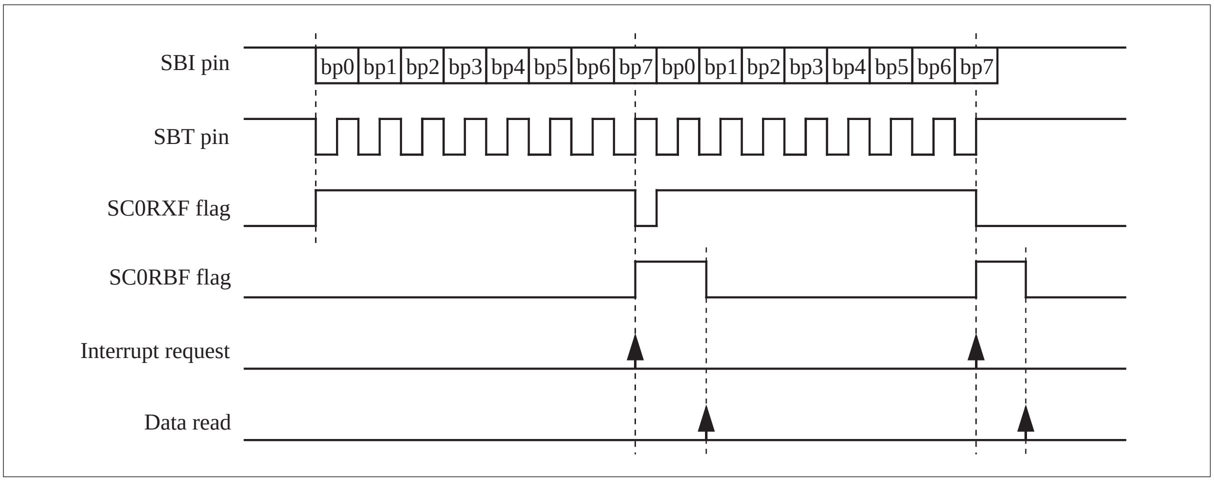


Fig. 13-2-6 Timing Chart (4)

After reception end (when the SC0RBF flag is "1"), the received data is fetched by reading the SC0RXB.
In the case of a 7-bit transfer, the MSB (bit 7) is "0".

The SC0RXF flag is set to "1" at the start of reception (at the falling edge of SBT pin), and is set to "0" at the end of reception.
The SC0RBF flag is set to "1" at the end of reception, and is set to "0" when SC0RXB is read.

Sending dummy data makes it possible to receive data while supplying the clock from the microcomputer side. In this case, interrupt requests are also generated by the transmission source. Receive data according to the following procedure:
(1) Select the internal clock as the reference clock, and set the parity, character length, etc.
(2) Enable both the transmission operation and the receiving operation.
(3) When dummy data is written to the transmission buffer, the clock is sent and reception begins.
When using clock synchronous mode (2), set the SBO pin as a general-purpose input port before writing the dummy data to the transmission buffer, and then reset the pin as the SBO pin after transmission is complete.

■ When a reception error is generated

- Transfer in clock synchronous mode with 8-bit data length, parity on.

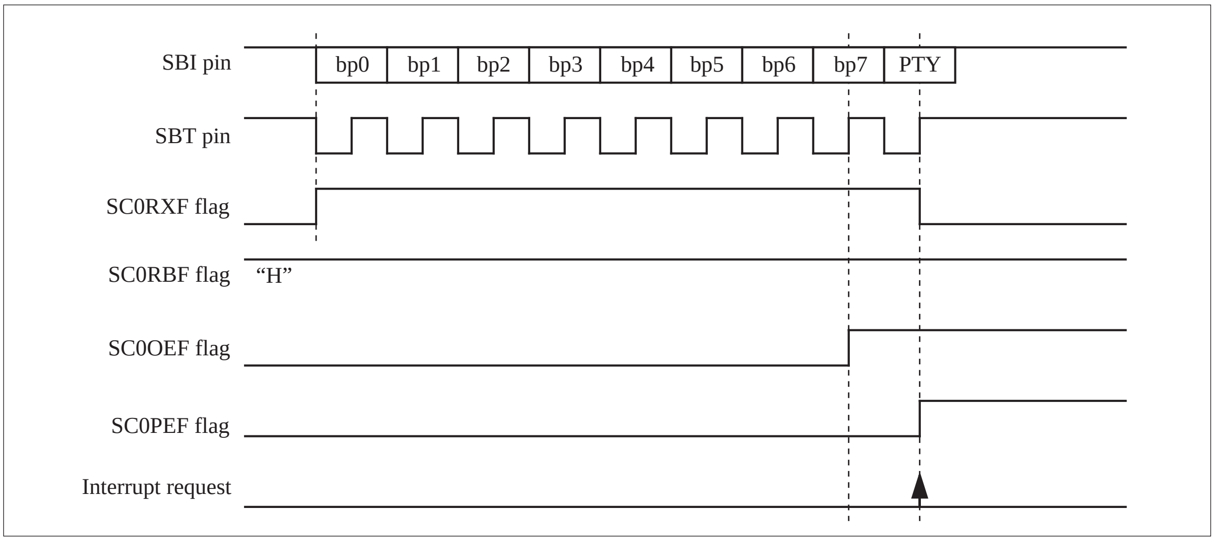


Fig. 13-2-7 Timing Chart (5)

When "reception end" is set as the reception interrupt source, an interrupt request is generated when reception ends, regardless of whether or not an error occurred.

When "reception end with error" is set as the reception interrupt source, an interrupt request is generated when reception ends with an error having occurred. (An interrupt request is not generated at the moment that the error occurred.)

An overrun error is generated when reception of the next data is completed before previously received data is read from the SC0RXB . In this event, the previously received data is lost. The overrun error indicator flag (SC0OEF) is updated at the moment the final data bit is received.

A parity error is generated when 0-fixed parity is set and a "1" is received, when 1-fixed parity is set and a "0" is received, when even parity is set and an odd number of ones is received, or when odd parity is set and an even number of ones is received. The parity error indicator flag (SC0PEF) is updated at the moment the parity bit is received.

<UART mode>

■ UART mode connection

Two different connection methods are possible, one for unidirectional transfer, and the other for bi-directional transfer.
The SBO pin is always an output, and the SBI pin is always an input.

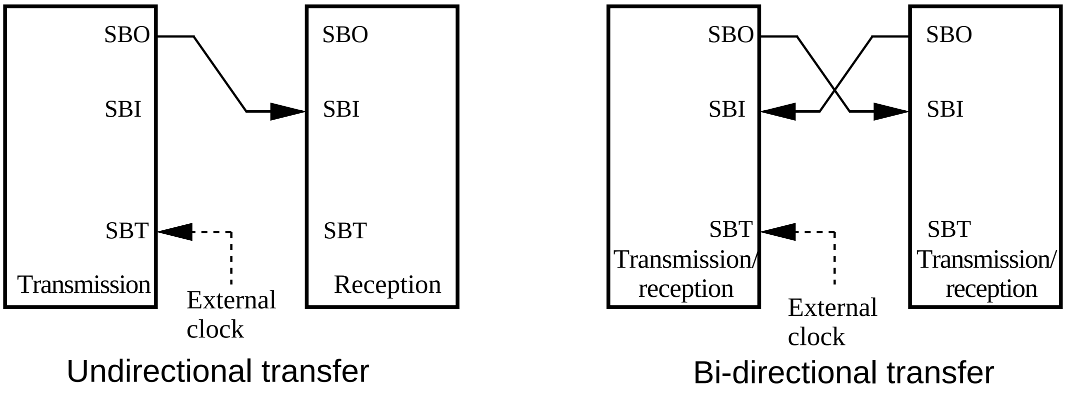


Fig. 13-2-8 Connections

■ UART mode bit rates

In UART mode, it is necessary to select an appropriate bit rate and serial interface input clock.
For example, when IOCLK = 15 MHz and the bit rate is 19.2 kbit/s, the timer function is used to divide the clock signal. The division ratio is determined as follows:

Timer division ratio = INT (IOCLK frequency/bit rate/8 + 0.5)

In the example described above, the timer division ratio is 98.
In the timer 3 base register, set TM3BR = 97 and set SC0CK2 to 0 = "100". IOCLK divided by 98 will then be supplied to the serial interface as the input clock. The bit rate error is calculated as follows:

Bit rate error = ABS (division ratio x 8 x bit rate/IOCLK frequency - 1)

In this example, the bit rate error is 0.35 %.
Typical examples are shown in Tables 13-2-2 through 13-2-4.

Note: When 1/8 of an external clock signal is used as the clock source, the high and low widths of the external clock must be at least 10, 5, or 2.5 SYSCLK cycles when (MCLK frequency/SYSCLK frequency) = 1, 2, or 4, respectively.

If the division ratio is large, use either a prescaler or a cascaded connection for the timer.

Table 13-2-2 Bit rates (1) (When IOCLK = 15 MHz)

Bit rate (bit/s)	When cascaded		When using prescalers	
	Timer division ratio	Bit rate error	Timer division ratio	Bit rate error
19 200	98	0.35 %	Not using	—
9 600	195	0.16 %	Not using	—
4 800	391	0.10 %	195 x 2	0.16 %
2 400	781	0.03 %	195 x 4	0.16 %
1 200	1 563	0.03 %	195 x 8	0.16 %

Table 13-2-3 Bit Rates (2) (When IOCLK = 12 MHz)

Bit rate (bit/s)	When cascaded		When using prescalers	
	Timer division ratio	Bit rate error	Timer division ratio	Bit rate error
19 200	78	0.16 %	Not using	—
9 600	156	0.16 %	Not using	—
4 800	313	0.16 %	156 x 2	0.16 %
2 400	625	0.0 %	125 x 5	0.0 %
1 200	1 250	0.0 %	125 x 10	0.0 %

Table 13-2-4 Bit Rates (3) (When IOCLK = 8 MHz)

Bit rate (bit/s)	When cascaded		When using prescalers	
	Timer division ratio	Bit rate error	Timer division ratio	Bit rate error
19 200	52	0.16 %	Not using	—
9 600	104	0.16 %	Not using	—
4 800	208	0.16 %	Not using	—
2 400	417	0.08 %	139 x 3	0.08 %
1 200	833	0.04 %	119 x 7	0.04 %

■ UART mode timing

<Transmission>

- Transfer with 8-bit data length, parity on, and 1 stop bit

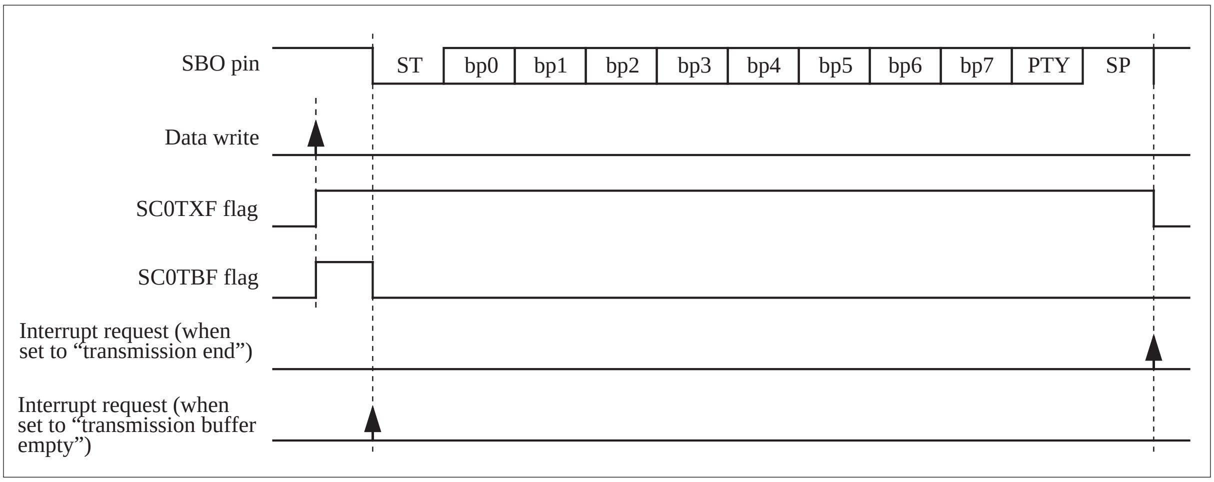

SBO pin
ST
bp0
bp1
bp2
bp3
bp4
bp5
bp6
bp7
PTY
SP
Data write
SC0TXF flag
SC0TBF flag
Interrupt request (when set to “transmission end”)
Interrupt request (when set to “transmission buffer empty”)

Fig. 13-2-9 Timing Chart (6)

- Two-byte transfer with 7-bit data length, parity on, and 1 stop bit

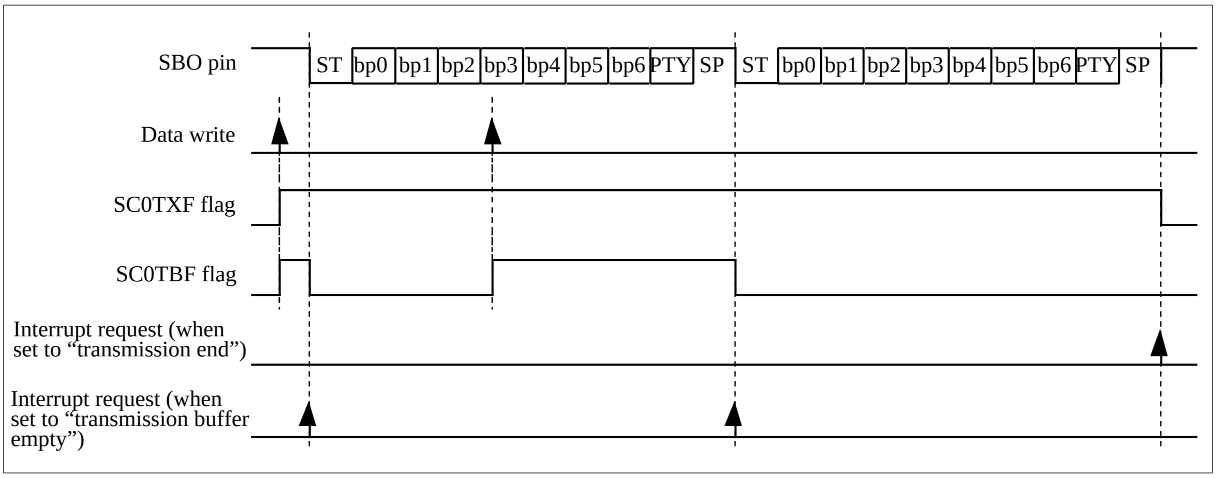

SBO pin
ST
bp0
bp1
bp2
bp3
bp4
bp5
bp6
PTY
SP
ST
bp0
bp1
bp2
bp3
bp4
bp5
bp6
PTY
SP
Data write
SC0TXF flag
SC0TBF flag
Interrupt request (when set to “transmission end”)
Interrupt request (when set to “transmission buffer empty”)

Fig. 13-2-10 Timing Chart (7)

When transmission is enabled, transmission starts when data is written to SC0TXB.
Continuous transmission is possible by writing data to SC0TXB again while transmission is in progress.
During a 7-bit transfer, the MSB (bit 7) is ignored.

The SC0TXF flag is set to "1" when data is written to SC0TXB, and is set to "0" at the end of transmission.
The SC0TBF flag is set to "1" when data is written to SC0TXB, and is set to "0" at the start of transmission.

<Reception>

- Transfer with 8-bit data length, parity on, and 1 stop bit

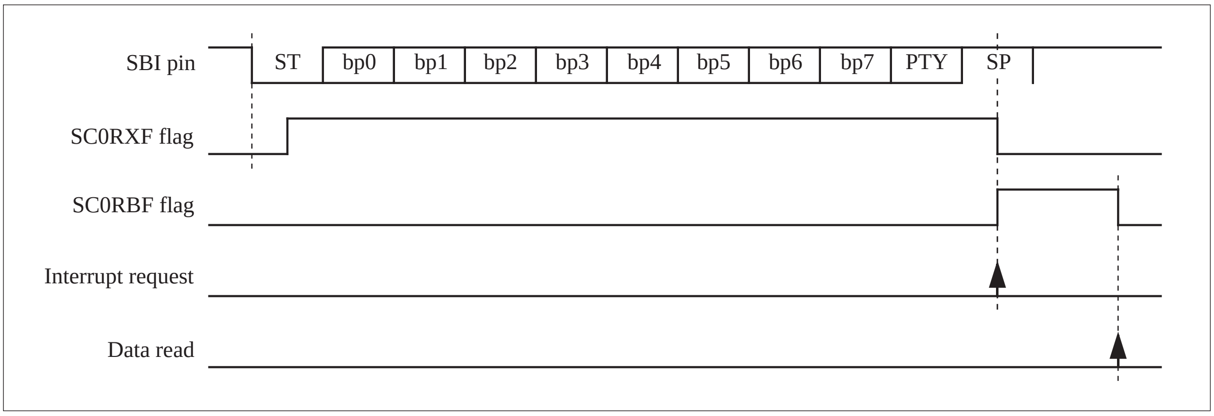


Fig. 13-2-11 Timing Chart (8)

- Two-byte transfer with 7-bit data length, parity on, and 1 stop bit

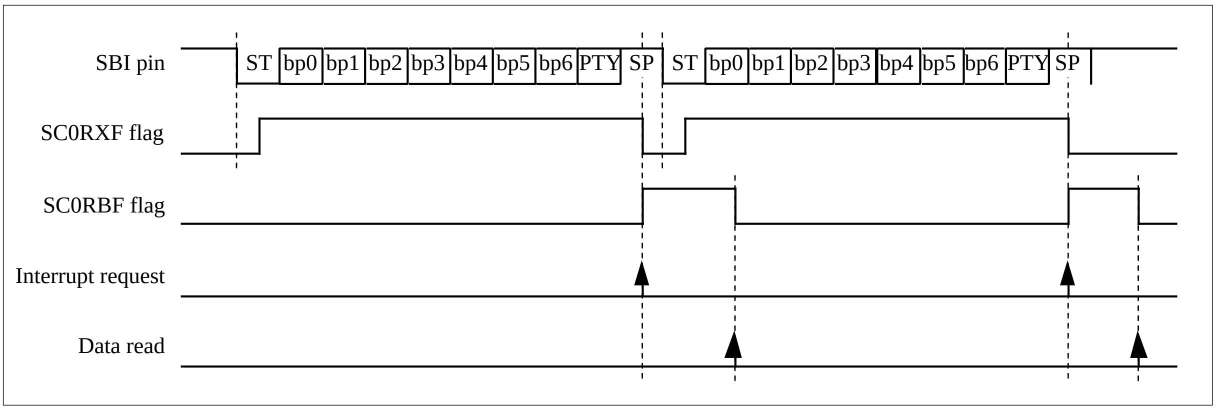


Fig. 13-2-12 Timing Chart (9)

After reception end (when the SC0RBF flag is "1"), the received data is fetched by reading the SC0RXB. In the case of a 7-bit transfer, the MSB (bit 7) is "0".

The SC0RXF flag is set to "1" at the start of reception (when the start bit is detected), and is set to "0" at the end of reception.
The SC0RBF flag is set to "1" at the end of reception, and is set to "0" when SC0RXB is read.

When a reception error is generated

- Transfer in UART mode with 8-bit data length, parity on, and 1 stop bit

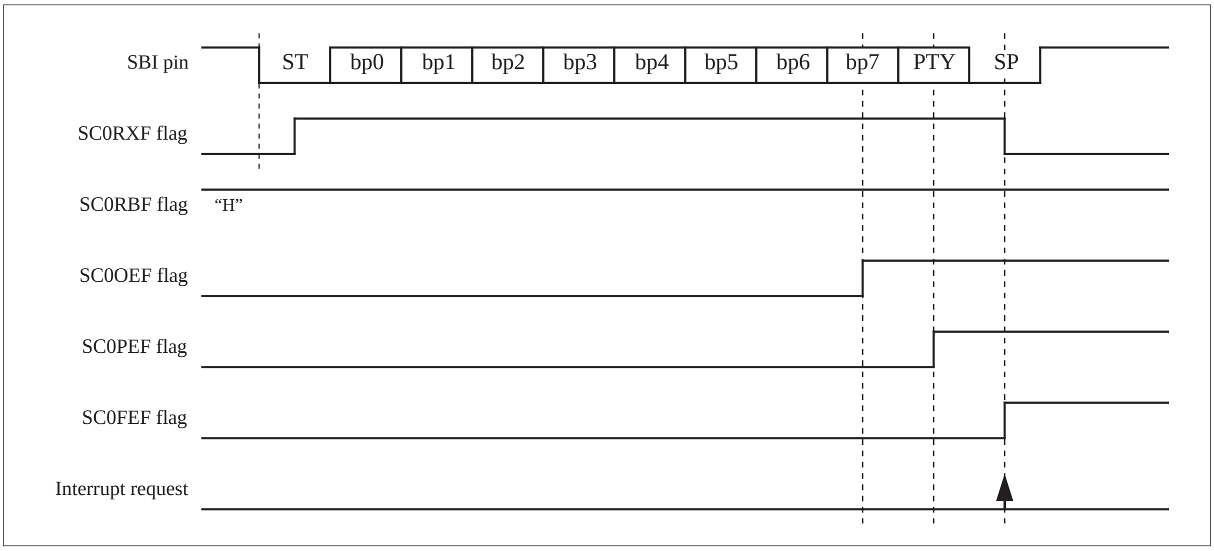


Fig. 13-2-13 Timing Chart (10)

When "reception end" is set as the reception interrupt source, an interrupt request is generated when reception ends, regardless of whether or not an error occurred.

When "reception end with error" is set as the reception interrupt source, an interrupt request is generated when reception ends with an error having occurred. (An interrupt request is not generated at the moment that the error occurred.)

An overrun error is generated when reception of the next data is completed before previously received data is read from the SC0RXB. In this event, the previously received data is lost. The overrun error indicator flag (SC0OEF) is updated at the moment the final data bit is received.

A parity error is generated when 0-fixed parity is set and a "1" is received, when 1-fixed parity is set and a "0" is received, when even parity is set and an odd number of ones is received, or when odd parity is set and an even number of ones is received. The parity error indicator flag (SC0PEF) is updated at the moment the parity bit are received.

A framing error is generated when "0" was received for the stop bit. The framing error indicator flag (SC0FEF) is updated at the moment the stop bit is received.

<I2C mode>

■ I2C mode connection

It is possible to connect a device that is capable of slave transmission and slave reception.
SDA and SCL require pull-up resistors. Connect pull-up resistors externally.
The SBO pin is an open-drain input/output, and the SBT pin is an open drain output.

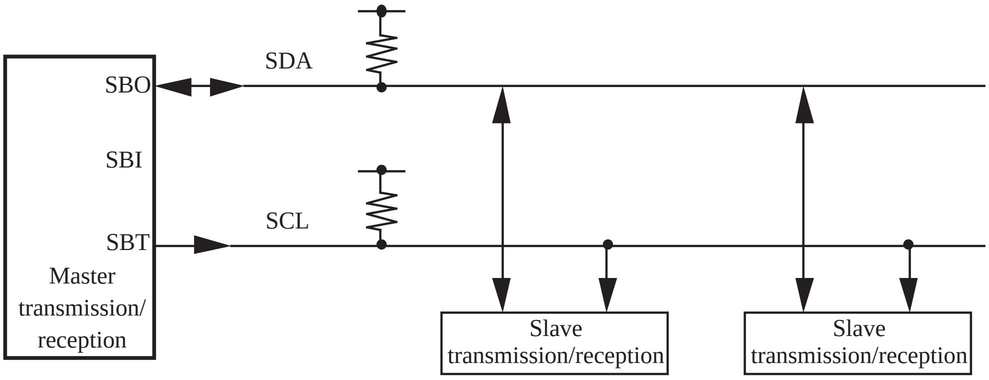


Fig.13-2-14 Connections

■ I2C mode transmission/reception

The transmission/reception procedure in I2C mode is described below.
(Refer to Fig. 13-2-15.)

- Make the initial settings as described below.

(1) I/O port setting
Set the SBT and SBO pins as general-purpose input ports.
For details on the settings, refer to the chapter on I/O ports.

(2) Transmission/reception mode setting (SC0CTR register)
Be certain to set the flags listed below to the specified values.

SBT0 pin output control flag (SC0TOE):	0
Protocol selection flags (SC0MD1,0):	10 (I2C mode)
I2C mode selection flag (SC0IIC):	0
Break transmission flag (SC0BKE):	0
Reception operation enable flag (SC0RXE):	0
Transmission operation enable flag (SC0TXE):	0

Flags other than those listed above may be set as desired.
However, the clock source must be selected from among the following four:
1/8 IOCLK
1/32 IOCLK
1/8 timer 3 underflow
1/8 timer 9 underflow
Set the parity bits each transmission/reception.

(3) I/O port setting
Set the I/O ports to SBT and SBO .
Leave the I/O port input/output control registers set to "input".

(4) Interrupt mode register setting (SC0ICR register)
Set the interrupt sources as "transmission end".

(5) Transmission/reception enable
Enable both transmission and reception operations.

- Send the start sequence (A) according to the procedure described below:

(1) Sending start sequence
When the I2C mode selection flag (SC0IIC) is changed from "0" to "1", a low signal is output on the SBO pin as the start sequence.

(2) Confirmation of sending start sequence
If the start sequence was generated normally, the I2C start sequence detection flag (SC0STF) changes to "1".
In this case, even if there are simultaneous starts, "arbitration lost" is not detected.

- Perform data transmission/reception (B) according to the procedure described below:

(1) Ack setting

"Ack" is represented by the parity bits.

Set the parity bit selection flags (SC0PB2 to 0) to "1 fixed" or "0 fixed" in accordance with the communications protocol for the device that is connected. (When sending "Ack", set "0 fixed"; when sending NO-Ack, set "1 fixed ".)

(2) Transmission/reception

When transmitting, write the data that is to be transmitted in the transmission buffer (SC0TXB).

When receiving, write "x'FF" in the transmission buffer (SC0TXB).

After the buffers are written, the clock signal is output and the transmission/reception operation is performed.

After transmission/reception are completed, the low signal output is maintained on the SBT and SBO pins.

(3) Reading the reception buffer

Always be certain to read the reception buffer after the transmission/reception operation is completed. (It is necessary to read the reception buffer after a transmission operation.)

(4) Ack confirmation

Ack can be read as a parity error. Read the parity error indication flag (SC0PEF). When parity is set to "0 fixed", the value of SC0PEF is the value of Ack.

When parity is set to "1 fixed", the inverted value of SC0PEF is the value of Ack.

When performing consecutive transmission/reception operations, repeat steps (1) to (4).

● Wait function under SCL control

The transmission/reception operation waits until SCL is released if SCL is driven low by a device performing slave transmission/reception. In this case, SCL goes high for 1/2 the normal interval (1/4 the one-bit width specified by the clock source selection flags (SC0CK2 to 0)).

- Send the stop sequence (C) according to the procedure described below:

(1) Sending stop sequence

When the I2C mode selection flag (SC0IIC) is changed from "1" to "0", the SBT pin goes high. One IOCLK cycle after the SBT pin goes high, the SBO pin goes high and the stop sequence is sent.

(2) Confirmation of sending stop sequence

If the stop sequence was generated normally, the I2C stop sequence detection flag (SC0SPF) changes to "1".

(3) Transmission/reception disable

Disable the transmission operation and the reception operation.

(Set SC0TXE and SC0RXE to "0". Be sure to always perform this step every time after the stop sequence is sent.)

If the above procedures do not satisfy the AC timing of the device that is connected, send the stop sequence according to the procedure described below.

(1)' SBT pin setting
Set the SBT pin as a general-purpose input port.
When the pin switches to a general-purpose input port, SCL goes high.

(2)' SBO pin setting
Set the SBO pin as a general-purpose input port.
When the pin switches to a general-purpose input port, SDA goes high and the stop sequence is generated.

(3)' Transmission/reception disable
Disable the transmission operation and the reception operation.
(Set SC0TXE and SC0RXE to "0". Be sure to always perform this step every time after the stop sequence is sent.)

(4)' I/O port setting
To perform further transmission/reception operations, set the I/O ports to SBT and SBO.

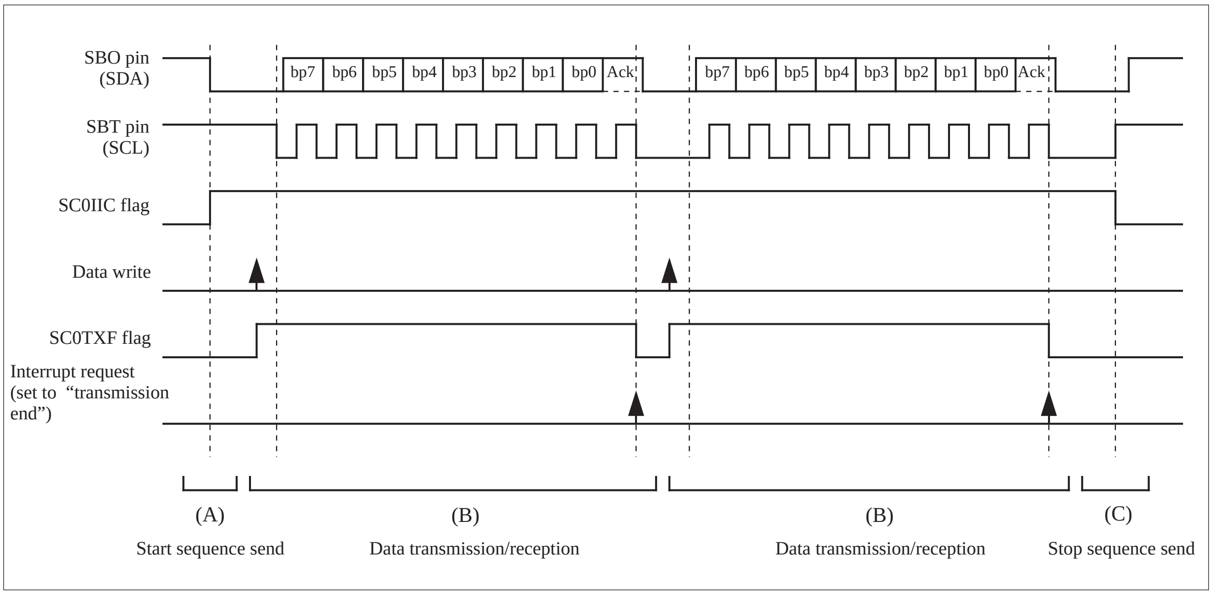


Fig. 13-2-15 Timing Chart (11)

- Resend the start sequence (D) according to the procedure described below.
  (Refer to Fig. 13-2-16.)

(1) SBO pin setting
Set the SBO pin as a general-purpose input port.
When the pin switches to a general-purpose input port, SDA goes high.

(2) SBT pin setting
Set the SBT pin as a general-purpose input port.
When the pin switches to a general-purpose input port, SCL goes high.

(3) Control register setting
Disable the transmission operation and the reception operation.
(Set SC0TXE and SC0RXE to "0".)
Set the I2C mode selection flag to "0".

(4) I/O port setting
Set the I/O ports to SBT and SBO.

(5) Transmission/reception enable
Enable the transmission operation and the reception operation.
(Set SC0TXE and SC0RXE to "1".)

(6) Start sequence resend
Set the I2C mode selection flag to "1".
A low signal is output on the SBO pin, and the start sequence is sent.

At this point, normal transmission/reception is now possible.

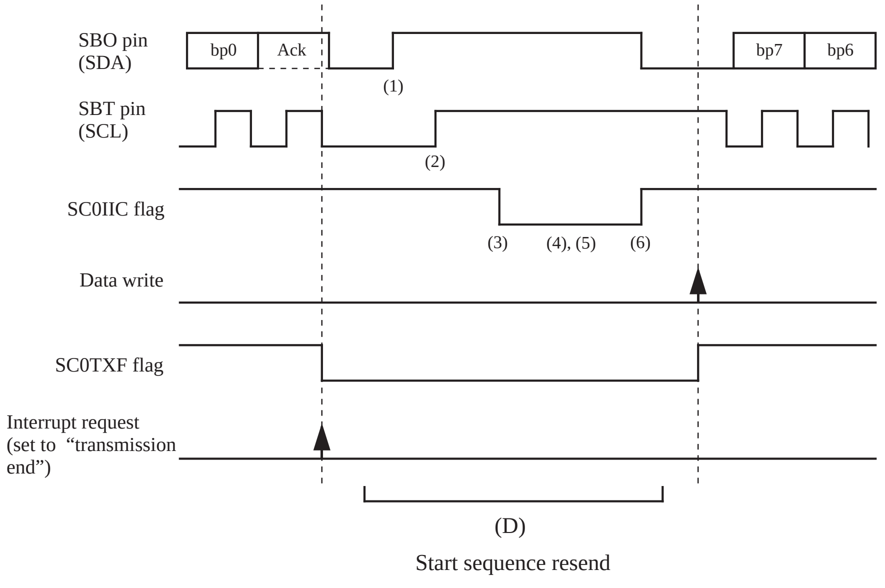


Fig. 13-2-16 Timing Chart (12)

13.3 Clock Synchronous Serial Interface

13.3.1 Features

Serial interfaces 1 and 2 are clock synchronous serial interfaces. Their features are described below.

- Parity None, 0 fixed, 1 fixed, even, odd
- Character length 7 bits, 8 bits
- Transmission and reception bit sequence
  LSB or MSB selectable
- Clock source 1/2, 1/8, or 1/32 of IOCLK (for either serial interface 1 or 2)
  1/8 of timer 2 or timer 8 underflow, 1/2 of timer 8 underflow
  (for either serial interface 1)
  1/8 of timer 3 or timer 9 underflow, 1/2 of timer 9 underflow
  (for either serial interface 2)
  External clock (for either serial interface 1 or 2)
- Maximum bit rate
  7.5 Mbit/s (when IOCLK is 15 MHz)
- Error detection during reception
  Parity errors, overrun errors
- Buffers Independent buffers for transmission and reception:
  Reception and transmission buffers are both double buffers
- Interrupts Transmission interrupts:
  "Transmission end" or "transmission buffer empty" selectable
  Reception interrupts:
  "Reception end" or "reception end with error" selectable

13.3.2 Block Diagram of Clock Synchronous Serial Interface

Fig 13-3-1 shows the block diagram for the clock synchronous serial interface sections.

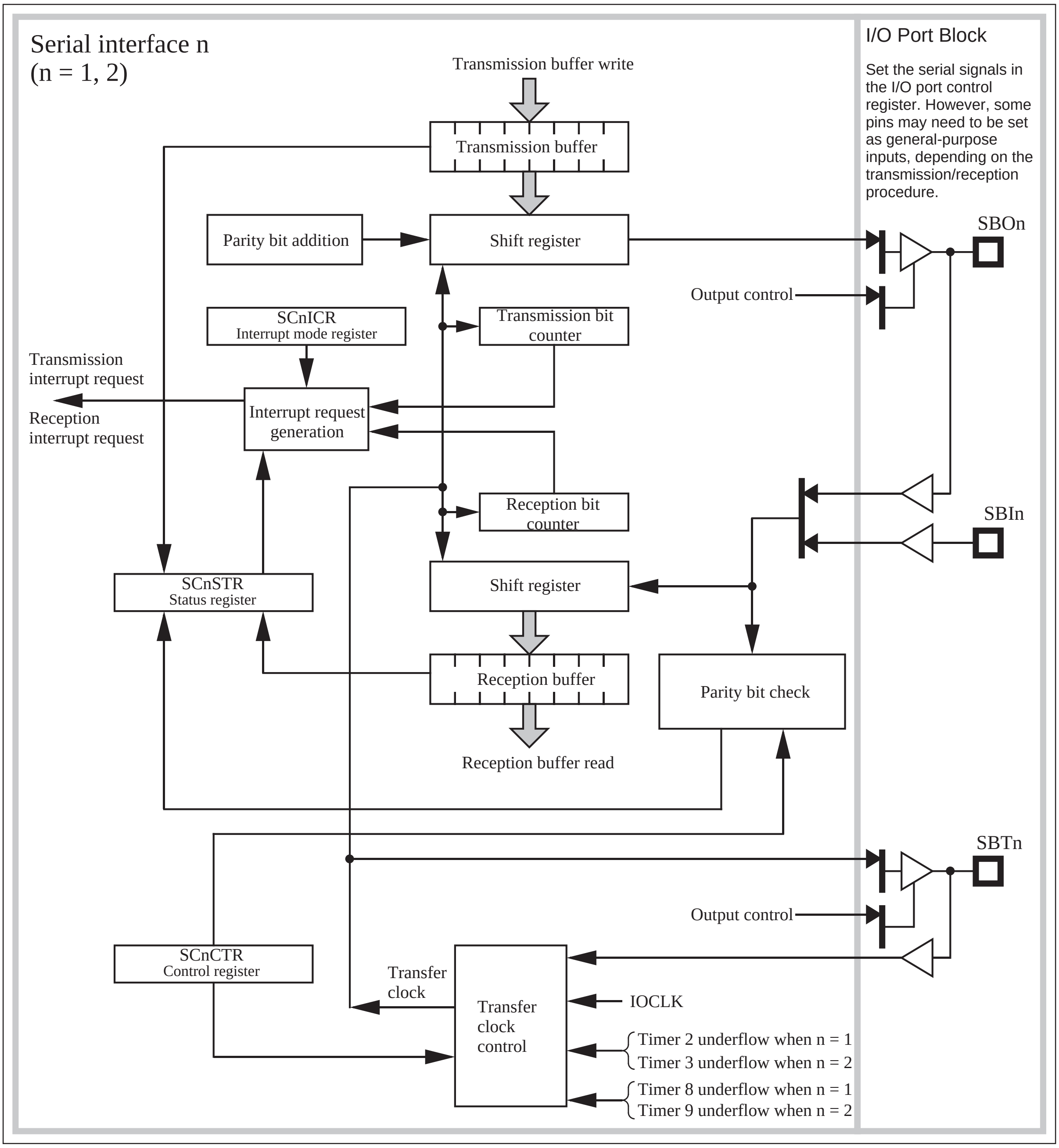

Serial interface n
(n = 1, 2)
Transmission buffer write
Transmission buffer
Parity bit addition
Shift register
SCnICR
Interrupt mode register
Transmission bit counter
Transmission interrupt request
Reception interrupt request
Interrupt request generation
Reception bit counter
SCnSTR
Status register
Shift register
Reception buffer
Parity bit check
Reception buffer read
SCnCTR
Control register
Transfer clock
Transfer clock control
IOCLK
Timer 2 underflow when n = 1
Timer 3 underflow when n = 2
Timer 8 underflow when n = 1
Timer 9 underflow when n = 2
I/O Port Block
Set the serial signals in the I/O port control register. However, some pins may need to be set as general-purpose inputs, depending on the transmission/reception procedure.
SBOn
Output control
SBIn
SBTn
Output control

Fig. 13-3-1 Block Diagram

13.3.3 Description of Registers for the Clock Synchronous Serial Interface

The clock synchronous serial interfaces include the registers listed in Table 13-3-1. These registers are used for settings such as clock source selection, parity bit selection, and protocol selection.

Table 13-3-1 List of Clock Synchronous Serial Interface Registers

Address	Name	Symbol	Number of bits	Initial value	Access size
x'34000810	Serial 1 control register	SC1CTR	16	x'0000	8, 16
x'34000814	Serial 1 interrupt mode register	SC1ICR	8	x'00	8
x'34000818	Serial 1 transmission buffer	SC1TXB	8	x'00	8
x'34000819	Serial 1 reception buffer	SC1RXB	8	x'00	8
x'3400081C	Serial 1 status register	SC1STR	8	x'00	8
x'34000820	Serial 2 control register	SC2CTR	16	x'0000	8, 16
x'34000824	Serial 2 interrupt mode register	SC2ICR	8	x'00	8
x'34000828	Serial 2 transmission buffer	SC2TXB	8	x'00	8
x'34000829	Serial 2 reception buffer	SC2RXB	8	x'00	8
x'3400082C	Serial 2 status register	SC2STR	8	x'00	8

Serial n control register (n = 1, 2)

Register symbol: SCnCTR

Address: x'34000810 (n =1), x'34000820 (n =2)

Purpose: This register sets the serial interface n operation control conditions.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	SCn TXE	SCn RXE	–	–	–	SCn MD0	SCn OD	SCn TOE	SCn CLN	SCn PB2	SCn PB1	SCn PB0	–	SCn CK2	SCn CK1	SCn CK0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit No.	Bit name	Description
0	SCnCK0	Clock source selection (LSB)
1	SCnCK1	Clock source selection
2	SCnCK2	Clock source selection (MSB) 000: 1/2 IOCLK 001: 1/8 IOCLK 010: 1/32 IOCLK 011: 1/2 timer 8 underflow (When n=1) 1/2 timer 9 underflow (When n=2) 100: 1/8 timer 2 underflow (When n=1) 1/8 timer 3 underflow (When n=2) 101: 1/8 timer 8 underflow (When n=1) 1/8 timer 9 underflow (When n=2) 110: Setting prohibited 111: External clock
3	—	"0" is returned when this bit is read.
4	SCnPB0	Parity bit selection (LSB)
5	SCnPB1	Parity bit selection
6	SCnPB2	Parity bit selection (MSB) 000: None 001, 010, 011: Setting prohibited 100: 0 fixed 101: 1 fixed 110: Even (even number of ones) 111: Odd (odd number of ones)
7	SCnCLN	Character length selection 0: 7 bits 1: 8 bits

Bit No.	Bit name	Description
8	SCnTOE	SBTn pin output control 0: When the internal clock is selected, the SBTn pin is an output only while transmission is in progress (the SBTn pin is an input when in standby mode or when an external clock is selected) 1: When the internal clock is selected, the SBTn pin is always an output (the SBTn pin is an input when an external clock is selected)
9	SCnOD	Transmission and reception bit sequence selection 0: From LSB 1: From MSB
10	SCnMD0	Protocol selection 0: Clock synchronous mode (1) (the SBOn pin is used as a data output, and the SBIn pin is used as a data input) 1: Clock synchronous mode (2) (the SBOn pin is used as a data input and output, and input on the SBIn pin is ignored)
13 to 11	—	"0" is returned when these bits are read.
14	SCnRXE	Reception operation enable 0: Disabled 1: Enabled
15	SCnTXE	Transmission operation enable 0: Disabled 1: Enabled

Serial n interrupt mode register (n = 1, 2)

Register symbol: SCnICR
Address: x'34000814 (n = 1), x'34000824 (n = 2)
Purpose: This register selects the sources for transmission interrupts and reception interrupts for serial interface n.

Bit No.	7	6	5	4	3	2	1	0
Bit name	SCn DMD	–	–	SCn TI	–	SCn RES	–	SCn RI
Reset	0	0	0	0	0	0	0	0
Access	R/W	R	R/W	R/W	R	R/W	R	R/W

Bit No.	Bit name	Description
0	SCnRI	Reception interrupt factor selection 0: Reception end 1: Reception end with error
1	—	"0" is returned when this bit is read.
2	SCnRES	Reception error interrupt factor selection 0: Interrupt request when an overrun or parity error occurs 1: Interrupt request when a parity error occurs
3	—	"0" is returned when this bit is read.
4	SCnTI	Transmission interrupt factor selection 0: Transmission end 1: Transmission buffer empty
5	Reserved	Setting "1" is prohibited.
6	—	"0" is returned when this bit is read.
7	SCnDMD	Data output retained during external clock transmission (valid only in clock synchronous mode) 0: Set data pin "H" at end of transmission 1: Maintain data pin at end of transmission

Serial n transmission buffer (n = 1, 2)

Register symbol SCnTXB
Address: x'34000818 (n=1), x'34000828 (n=2)
Purpose: This register writes the transmission data to serial interface n.

Bit No.	7	6	5	4	3	2	1	0
Bit name	SCn TXB7	SCn TXB6	SCn TXB5	SCn TXB4	SCn TXB3	SCn TXB2	SCn TXB1	SCn TXB0
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Data is transmitted by writing it to this buffer.

Serial n reception buffer (n = 1, 2)

Register symbol: SCnRXB
Address: x'34000819 (n=1), x'34000829(n=2)
Purpose: This register reads in the reception data of serial interface n.

Bit No.	7	6	5	4	3	2	1	0
Bit name	SCn RXB7	SCn RXB6	SCn RXB5	SCn RXB4	SC0n RXB3	SCn RXB2	SCn RXB1	SCn RXB0
Reset	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R

Reception data is gotten by reading this buffer at the end of reception.
In the case of a 7-bit transfer, the MSB (bit 7) is "0".

Serial n status register (n=1,2)

Register symbol: SCnSTR

Address: x'3400081C (n=1), x'3400082C (n=2)

Purpose: This register indicates the status of serial interface n.

Bit No.	7	6	5	4	3	2	1	0
Bit name	SCn TXF	SCn RXF	SCn TBF	SCn RBF	–	–	SCn PEF	SCn OEF
Reset	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R

Bit No.	Bit name	Description
0	SCnOEF	Overrun error indication 0: No error 1: Overrun error occurred
1	SCnPEF	Parity error indication 0: No error 1: Parity error occurred
3 to 2	—	"0" is returned when these bits are read.
4	SCnRBF	Reception buffer status indication 0: Reception buffer empty 1: Data exists in the reception buffer
5	SCnTBF	Transmission buffer status indication 0: Transmission buffer empty 1: Data exists in the transmission buffer
6	SCnRXF	Reception status indication 0: Waiting for reception 1: Reception in progress
7	SCnTXF	Transmission status indication 0: Ready for transmission 1: Transmission in progress

13.3.4 Description of Operation

■ Clock synchronous serial interface n connection (n=1,2)

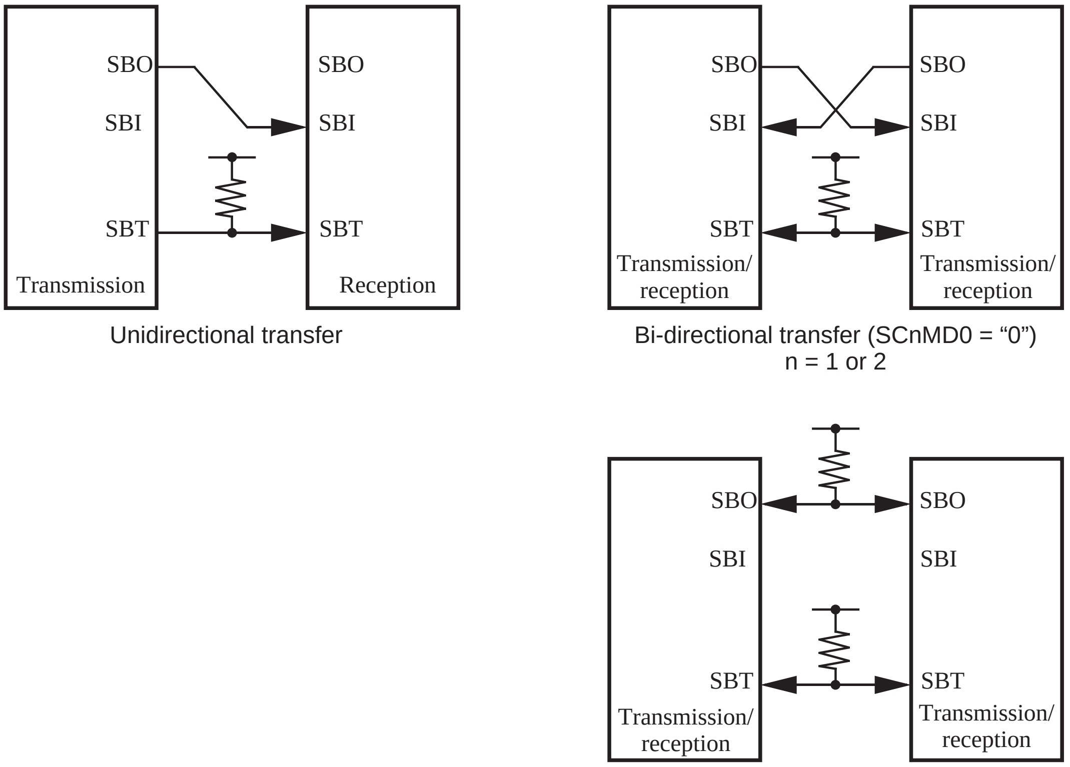
Two different connection methods are possible, one for unidirectional transfer, and the other for bi-directional transfer.

When SBT pin is an output only during transmission (SCnTOE = "0"), it is necessary to pull up SBT pin. In addition, when using SBO pin as a data input/output (SCnMD0 = "1"), it is necessary to pull up SBO pin. Connect a pull-up resistor externally.

When using SBO pin as a data output and SBI pin as a data input (SCnMD0 = "0"), the SBO pin is always an output and the SBI pin is always an input.

When using SBO pin as a data input/output (SCnMD0 = "1"), the SBO pin is an output only during transmission, and is normally an input.

When SCnTOE is "0", the SBT pin is an output only during transmission with the internal clock, and is normally an input. Furthermore, when SCnTOE is "1", the SBT pin is always an output when the internal clock is selected.



Unidirectional transfer

Bi-directional transfer (SCnMD0 = “0”)
n = 1 or 2

Bi-directional transfer (SCnMD0 = “1”)
n = 1 or 2

Fig. 13-3-2 Connections

■ Clock synchronous serial interface timing

<Transmission>

- One-byte transfer with 8-bit data length and parity off

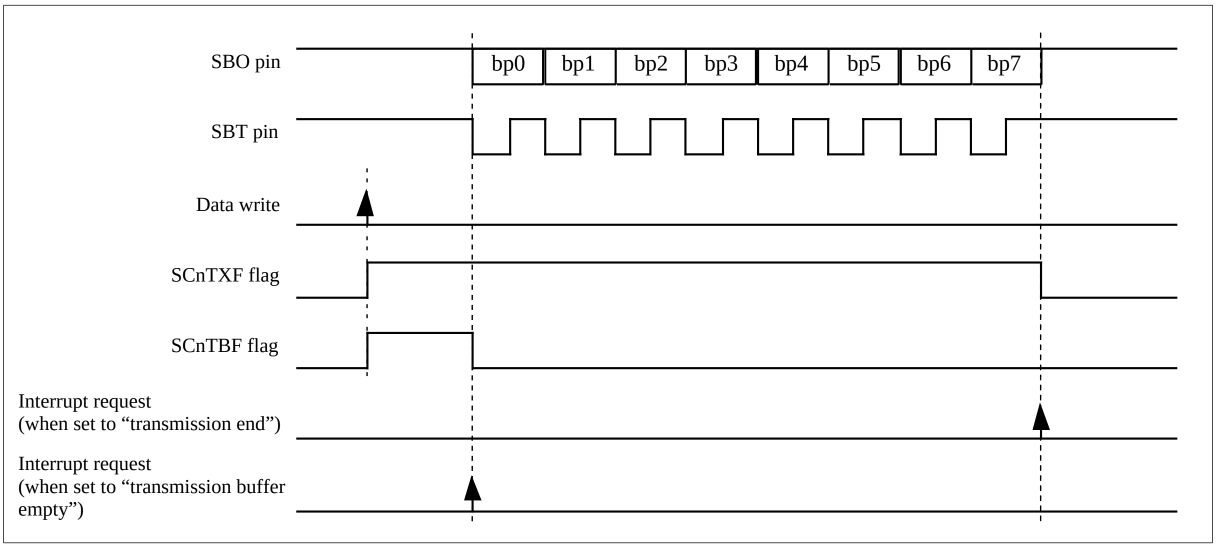


Fig. 13-3-3 Timing Chart (13)

- Two-byte transfer with 7-bit data length and parity on

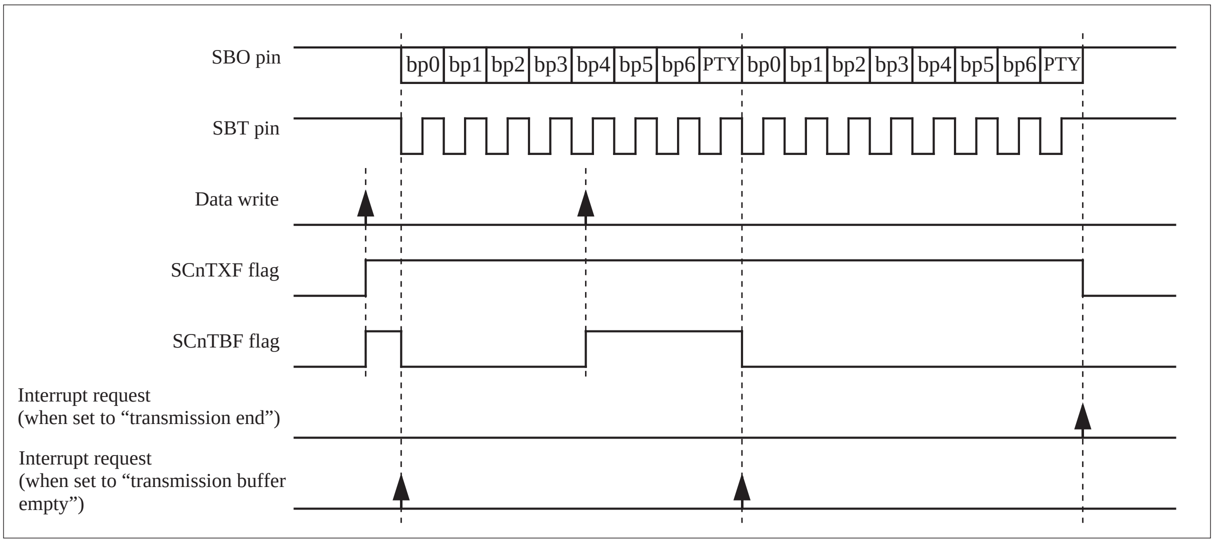


Fig. 13-3-4 Timing Chart (14)

When transmission is enabled, transmission starts when data is written to SCnTXB.
Continuous transmission is possible by writing data to SCnTXB again while transmission is in progress.
During a 7-bit transfer, the MSB (bit 7) is ignored.

The SCnTXF flag is set to "1" when data is written to SCnTXB, and is set to "0" at the end of transmission.
The SCnTBF flag is set to "1" when data is written to SCnTXB, and is set to "0" at the start of transmission.

<Reception>

- One-byte transfer with 7-bit data length and parity on

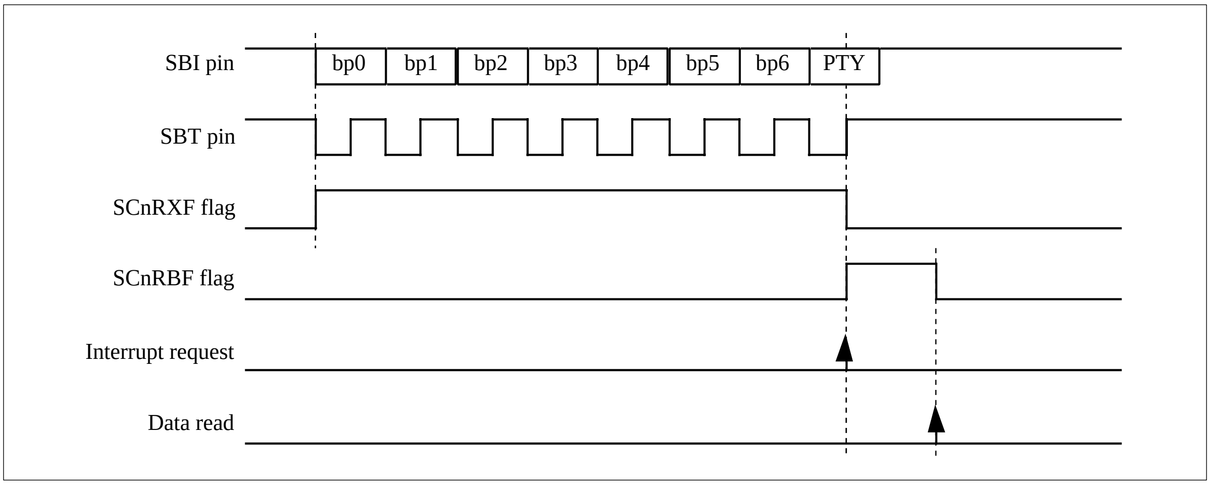


Fig. 13-3-5 Timing Chart (15)

- Two-byte transfer with 8-bit data length and parity on

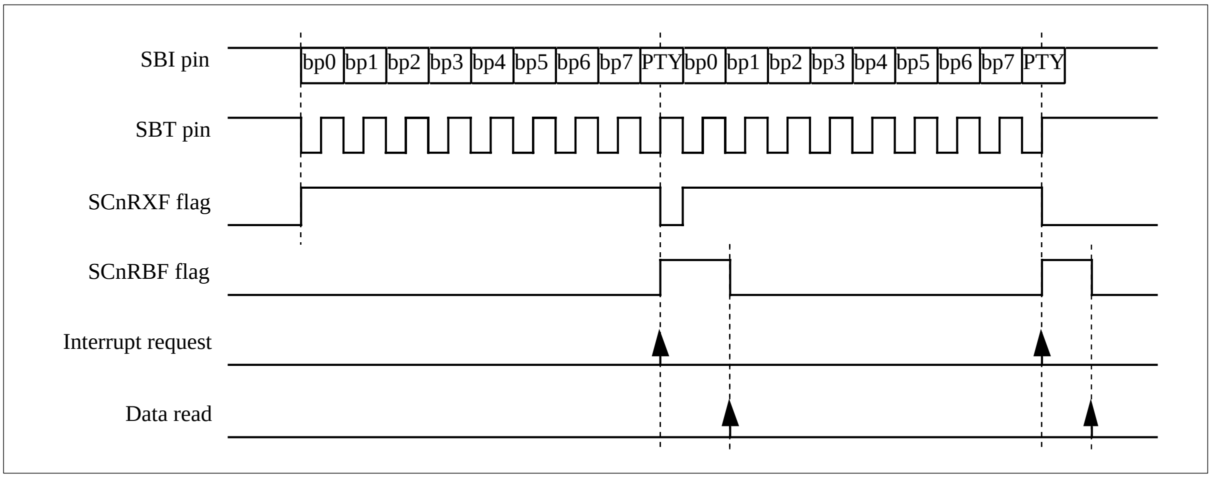


Fig. 13-3-6 Timing Chart (16)

After reception end (when the SCnRBF flag = “1”), the received data is fetched by reading the SCnRXB. In the case of a 7-bit transfer, the MSB (bit 7) is “0”.

The SCnRXF flag is set to “1” at the start of reception (at the falling edge of SBT pin), and is set to “0” at the end of reception.
The SCnRBF flag is set to “1” at the end of reception, and is set to “0” when SCnRXB is read.

Sending dummy data makes it possible to receive data while supplying the clock from the microcomputer side. In this case, interrupt requests are also generated by the transmission source. Receive data according to the following procedure:
(1) Select the internal clock as the reference clock, and set the parity, character length, etc.
(2) Enable both the transmission operation and the receiving operation.
(3) When dummy data is written to the transmission buffer, the clock is sent and reception begins.
When using clock synchronous mode (2), set the SBO pin as a general-purpose input port before writing the dummy data to the transmission buffer, and then reset the pin as the SBO pin after transmission is complete.

■ When a reception error is generated

- Transfer with 7-bit data length, parity on

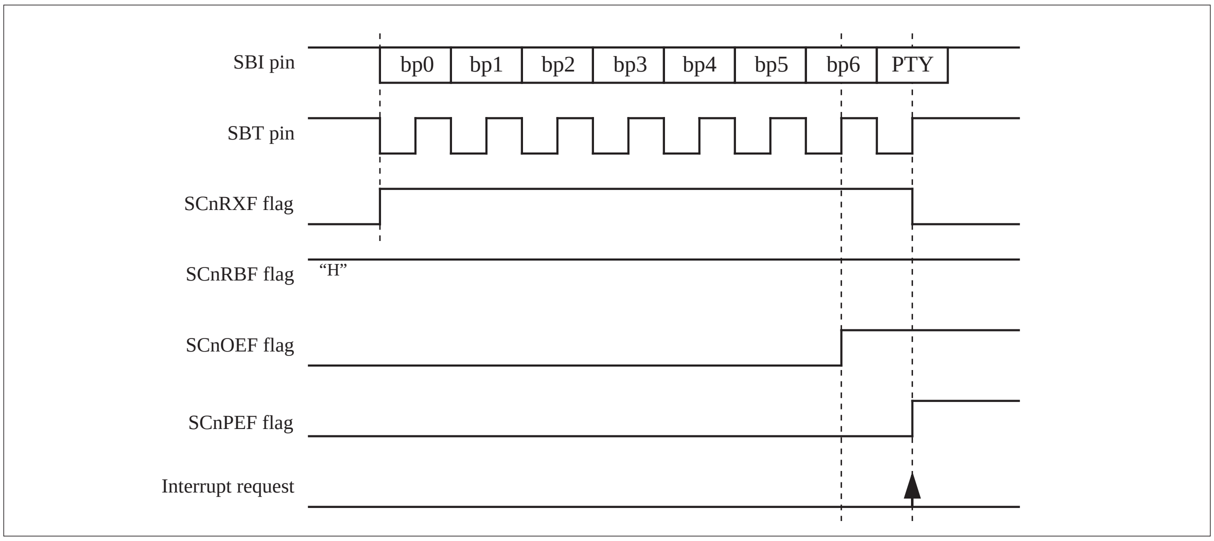


Fig. 13-3-7 Timing Chart (17)

When "reception end" is set as the reception interrupt source, an interrupt request is generated when reception ends, regardless of whether or not an error occurred.

When "reception end with error" is set as the reception interrupt source, an interrupt request is generated when reception ends with an error having occurred. (An interrupt request is not generated at the moment that the error occurred.)

An overrun error is generated when reception of the next data is completed before previously received data is read from the SCnRXB. In this event, the previously received data is lost. The overrun error indicator flag (SCnOEF) is updated at the moment the final data bit is received.

A parity error is generated when 0-fixed parity is set and a "1" is received, when 1-fixed parity is set and a "0" is received, when even parity is set and an odd number of ones is received, or when odd parity is set and an even number of ones is received. The parity error indicator flag (SCnPEF) is updated at the moment the parity bit is received.

13.4 Universal Asynchronous Receiver-Transceiver Serial Interface

13.4.1 Features

Serial interface 3 is a UART serial interface. Its features are described below.

- Parity None, 0 fixed, 1 fixed, even, odd
- Character length 7 bits, 8 bits
- Transmission and reception bit sequence
  LSB or MSB selectable
- Clock source IOCLK
  Timer 2 or timer 8 underflow
  External clock
- Dedicated counter built in
  Has an internal 7-bit counter that permits fast bit rates even with a clock source that operates at a comparatively low frequency.
- Maximum bit rate
  230.4 kbit/s (when IOCLK = 15 MHz)
- Error detection during reception
  Parity errors, overrun errors, flaming error
- Buffers Independent buffers for transmission and reception:
  Reception and transmission buffers are both double buffers
- Interrupts Transmission interrupts:
  "Transmission end" or "transmission buffer empty" selectable
  Reception interrupts:
  "Reception end" or "reception end with error" selectable
- Transmission interrupt function
  Transmissions can be interrupted and resumed with the proper external pin and register settings.

13.4.2 Block Diagram of UART Serial Interface

Fig 13-4-1 shows the block diagram for the UART serial interface sections.

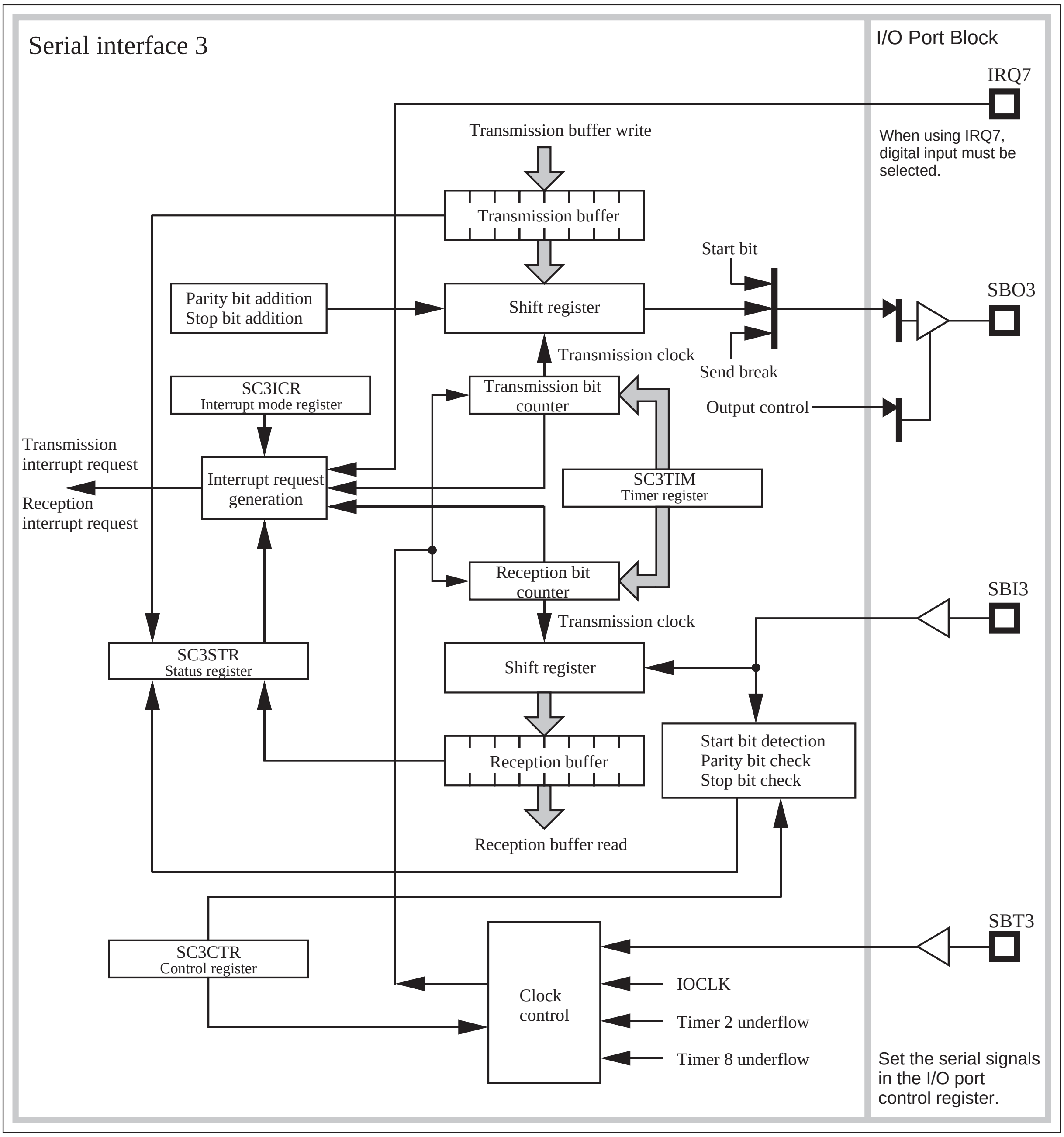


Fig. 13-4-1 Block Diagram

13.4.3 Description of Registers for the UART Serial Interface

The UART serial interface includes the registers listed in Table 13-4-1. These registers are used for settings such as clock source selection, parity bit selection, and protocol selection.

Table 13-4-1 List of UART Serial Interface Registers

Address	Name	Symbol	Number of bits	Initial value	Access size
x'34000830	Serial 3 control register	SC3CTR	16	x'0000	8, 16
x'34000834	Serial 3 interrupt mode register	SC3ICR	8	x'00	8
x'34000838	Serial 3 transmission buffer	SC3TXB	8	x'00	8
x'34000839	Serial 3 reception buffer	SC3RXB	8	x'00	8
x'3400083C	Serial 3 status register	SC3STR	8	x'00	8
x'3400083D	Serial 3 timer register	SC3TIM	8	x'00	8

Serial 3 control register

Register symbol: SC3CTR
Address: x'34000830
Purpose: This register sets the serial interface 3 operation control conditions.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	SC3 TXE	SC3 RXE	SC3 BKE	SC3 TWS	–	–	SC3 OD	SC3 TWE	SC3 CLN	SC3 PB2	SC3 PB1	SC3 PB0	SC3 STB	–	SC3 CK1	SC3 CK0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W

Bit No.	Bit name	Description
0	SC3CK0	Clock source selection (LSB)
1	SC3CK1	Clock source selection (MSB) 00: IOCLK 01: Timer 2 underflow 10: External clock 11: Timer 8 underflow
2	—	"0" is returned when this bit is read.
3	SC3STB	Stop bit selection 0: 1 bit 1: 2 bits
4	SC3PB0	Parity bit selection (LSB)
5	SC3PB1	Parity bit selection
6	SC3PB2	Parity bit selection (MSB) 000: None 001, 010, 011: Setting prohibited 100: 0 fixed 101: 1 fixed 110: Even (even number of ones) 111: Odd (odd number of ones)
7	SC3CLN	Character length selection 0: 7 bits 1: 8 bits

Bit No.	Bit name	Description
8	SC3TWE	Transmission interrupt enable 0: Interrupt disable 1: Interrupt enable
9	SC3OD	Transmission and reception bit sequence selection 0: From LSB 1: From MSB
11 to 10	—	"0" is returned when this bit is read
12	SC3TWS	Transmission interrupt code selection 0: Interrupt when external pin IRQ7 is “L” 1: Interrupt when external pin IRQ7 is “H”
13	SC3BKE	Break transmission (SBO3 pin is fixed at "0") 0: Normal operation 1: Send break
14	SC3RXE	Reception operation enable 0: Disabled 1: Enabled
15	SC3TXE	Transmission operation enable 0: Disabled 1: Enabled

Serial 3 interrupt mode register

Register symbol: SC3ICR

Address: x'34000834

Purpose: This register selects the sources for transmission interrupts and reception interrupts for serial interface 3.

Bit No.	7	6	5	4	3	2	1	0
Bit name	–	–	–	SC3 TI	–	SC3 RES	–	SC3 RI
Reset	0	0	0	0	0	0	0	0
Access	R	R	R/W	R/W	R	R/W	R	R/W

Bit No.	Bit name	Description
0	SC3RI	Reception interrupt factor selection 0: Reception end 1: Reception end with error
1	—	"0" is returned when this bit is read
2	SC3RES	Reception error interrupt factor selection 0: Interrupt request when an overrun, parity, or framing error occurs 1: Interrupt request when a parity error occurs
3	—	"0" is returned when this bit is read
4	SC3TI	Transmission interrupt factor selection 0: Transmission end 1: Transmission buffer empty
5	Reserved	Setting "1" is prohibited.
7 to 6	—	"0" is returned when these bits are read

Serial 3 transmission buffer

Register symbol: SC3TXB
Address: x'34000838
Purpose: This register writes the transmission data of serial interface 3.

Bit No.	7	6	5	4	3	2	1	0
Bit name	SC3 TXB7	SC3 TXB6	SC3 TXB5	SC3 TXB4	SC3 TXB3	SC3 TXB2	SC3 TXB1	SC3 TXB0
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Data is transmitted by writing it to this buffer.

Serial 3 reception buffer

Register symbol: SC3RXB
Address: x'34000839
Purpose: This register reads in the reception data of serial interface 3.

Bit No.	7	6	5	4	3	2	1	0
Bit name	SC3 RXB7	SC3 RXB6	SC3 RXB5	SC3 RXB4	SC3 RXB3	SC3 RXB2	SC3 RXB1	SC3 RXB0
Reset	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R

Reception data is gotten by reading this buffer at the end of reception.
In the case of a 7-bit transfer, the MSB (bit 7) is “0”.

Serial 3 status register

Register symbol: SC3STR
Address: x'3400083C
Purpose: This register indicates the status of serial interface 3.

Bit No.	7	6	5	4	3	2	1	0
Bit name	SC3 TXF	SC3 RXF	SC3 TBF	SC3 RBF	SC3 CTS	SC3 FEF	SC3 PEF	SC3 OEF
Reset	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R

Bit No. Bit name Description

0 SC3OEF Overrun error indication
0: No error
1: Overrun error occurred

1 SC3PEF Parity error indication
0: No error
1: Parity error occurred

2 SC3FEF Framing error indication
0: No error
1: Framing error occurred

3 SC3CTS External pin IRQ7 status indication
"0" when IRQ7 is “L”, and "1" when “H”.
Note: When P83A of the port 8 analog/digital input control register P8AD is “1”, IRQ7 is treated as “L” internally by the microcontroller and reading the SC3CTS bit returns a value of “0”, regardless of the actual values of the port pins.

4 SC3RBF Reception buffer status indication
0: Reception buffer empty
1: Data exists in the reception buffer

5 SC3TBF Transmission buffer status indication
0: Transmission buffer empty
1: Data exists in the transmission buffer

6 SC3RXF Reception status indication
0: Waiting for reception
1: Reception in progress

7 SC3TXF Transmission status indication
0: Ready for transmission
1: Transmission in progress

Serial 3 timer register

Register symbol: SC3TIM
Address: x'3400083D
Purpose: This register sets the timer that is used for internal division for serial interface 3.

Bit No.	7	6	5	4	3	2	1	0
Bit name	–	SC3 TIM6	SC3 TIM5	SC3 TIM4	SC3 TIM3	SC3 TIM2	SC3 TIM1	SC3 TIM0
Reset	0	0	0	0	0	0	0	0
Access	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Set the value that corresponds to the required division ratio - 1.

13.4.4 Description of Operation

■ UART Serial Interface connection

Two different connection methods are possible, one for unidirectional transfer, and the other for bi-directional transfer.

The SBO pin is always an output, and the SBI pin is always an input.

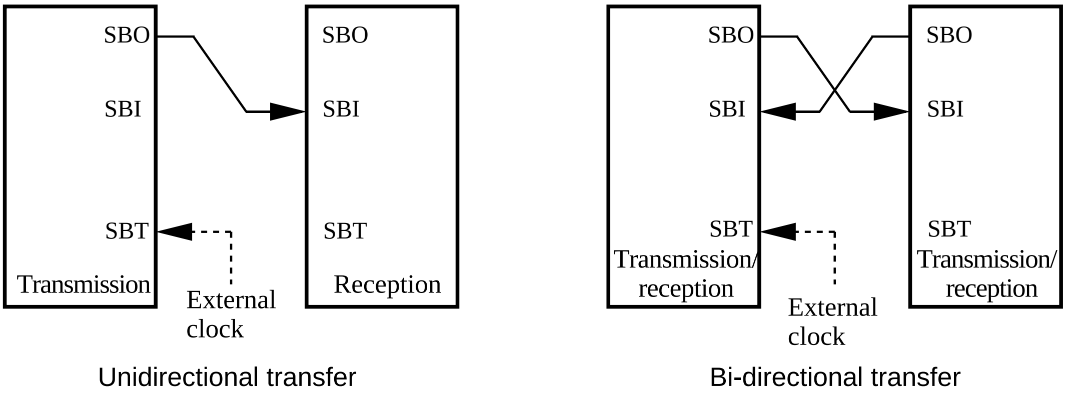


Fig. 13-4-2 Connections

■ UART serial interface bit rate

The UART serial interface has an internal 7-bit dedicated counter that supports fast bit rates even with a comparatively slow clock source.

For example, when IOCLK is being used and a transfer is being performed, the division ratio should be set as follows:

Note: When an external clock signal is used as the clock source, the high and low widths of the external clock must be at least 10, 5, or 2.5 SYSCLK cycles when (MCLK frequency/SYSCLK frequency) = 1, 2, or 4, respectively.

Division ratio 1 = INT (IOCLK frequency / bit rate/127) + 1
Division ratio 2 = INT (IOCLK frequency / bit rate/division ratio 1 + 0.5)
Subtract 1 from the value for division ratio 2 that was derived through the above equations, and write the result in SC3TIM.
If the value of division ratio 1 is 2 or higher, timer 2 or timer 8 must be used to divide the clock. Set SC3CK1 and 0 in the serial 3 control register SC3CTR to "01", and then set the control registers of timer 2 so that the clock is divided by the value of division ratio 1. (Otherwise, set SC3CK1 and 0 in the serial 3 control register SC3CTR to "11", and then set the control registers of timer 8 so that the clock is divided by the value of division ratio 1.)
If the value of division ratio 1 is 1, set SC3CK1 and 0 in the serial 3 control register SC3CTR to "00" and select IOCLK.
The error versus the actual bit rate is calculated as follows:
Bit rate error = ABS (division ratio 1 x division ratio 2 x bit rate / IOCLK frequency – 1)

For example, when a 15 MHz IOCLK signal is used and transfer is conducted at a rate of 38.4 kbit/s, the timer function is used to divide the clock signal. According to the equations shown above, division ratio 1 is 4 and division ratio 2 is 98.
Set TM2BR = 3 in the timer 2 base register and SC3TIM = 97 in the serial 3 timer register, and set SC3CK1 and 0 to "01".
According to the equation shown above, the bit rate error is 0.35 %.

Tables 13-4-2 through 13-4-4 show typical examples.

Table 13-4-2 Bit Rates (1) (When IOCLK = 15 MHz)

Bit rate (bit/s)	Division ratio 1	Division ratio 2	Bit rate error
230 400	1	65	0.16 %
115 200	2	65	0.16 %
56 000	3	89	0.32 %
38 400	4	98	0.35 %
19 200	7	112	0.35 %
9 600	13	120	0.16 %
4 800	25	125	0.00 %
2 400	50	125	0.00 %
1 200	99	126	0.21 %
600	197	127	0.08 %
300	394	127	0.08 %
150	788	127	0.08 %

Note: When using a timer to divide the clock signal, subtract 1 from the value of division ratio 1 as derived from the equations on the top of this page, and write the result in the timer base register. For details, refer to the chapter on the 8-bit timers.

Table 13-4-3   Bit Rates (2) (When IOCLK = 12 MHz)

Bit rate (bit/s)	Division ratio 1	Division ratio 2	Bit rate error
230 400	1	52	0.16 %
115 200	1	104	0.16 %
56 000	2	107	0.13 %
38 400	3	104	0.16 %
19 200	5	125	0.00 %
9 600	10	125	0.00 %
4 800	20	125	0.00 %
2 400	40	125	0.00 %
1 200	79	127	0.33 %
600	158	127	0.33 %
300	315	127	0.01 %
150	630	127	0.01 %

Table 13-4-4   Bit Rates (3) (When IOCLK = 8 MHz)

Bit rate (bit/s)	Division ratio 1	Division ratio 2	Bit rate error
230 400	1	35	0.80 %
115 200	1	69	0.64 %
56 000	2	71	0.60 %
38 400	2	104	0.16 %
19 200	4	104	0.16 %
9 600	7	119	0.04 %
4 800	14	119	0.04 %
2 400	27	123	0.37 %
1 200	53	126	0.17 %
600	105	127	0.01 %
300	210	127	0.01 %
150	420	127	0.01 %

[Notes on Usage]

1 Set SC3CTR before setting the other registers, and do not change the setting while transmitting or receiving, or while there is data in the transmission buffer. Operation is not guaranteed if the setting of the SC3CTR register is changed.

2 Before writing to the transmission buffer SC3TXB, confirm that the transmission buffer is empty. When writing, either first confirm that the transmission buffer is empty by checking SC3TBF in the SC3STR status register and then write the data, or else set SC3TI in the SC3ICR interrupt mode register to "1" and then write the data during the appropriate interrupt processing.

3 Set a value of 16 or higher in SC3TIM.

4 When using an external clock (SBT3 pin), the high and low widths of the external clock must be at least 10, 5, or 2.5 SYSCLK cycles when (MCLK frequency/SYSCLK frequency) = 1, 2, or 4, respectively.

■ UART Serial Interface timing

<Transmission>

- Transfer with 7-bit data length, parity off, and 2 stop bit

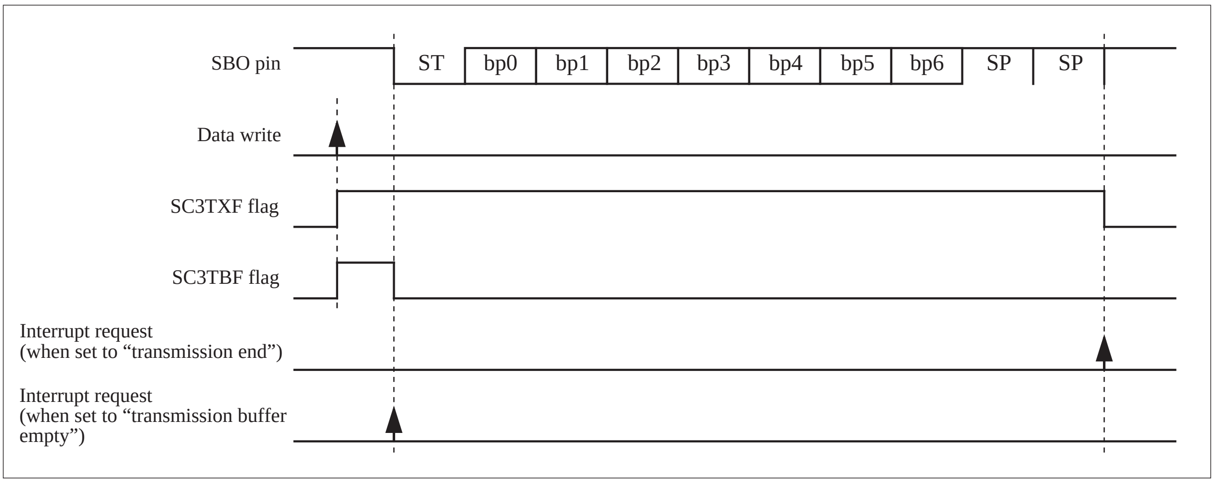

SBO pin
ST
bp0
bp1
bp2
bp3
bp4
bp5
bp6
SP
SP
Data write
SC3TXF flag
SC3TBF flag
Interrupt request (when set to “transmission end”)
Interrupt request (when set to “transmission buffer empty”)

Fig. 13-4-3 Timing Chart (18)

- Two-byte transfer with 8-bit data length, parity off, and 1 stop bit

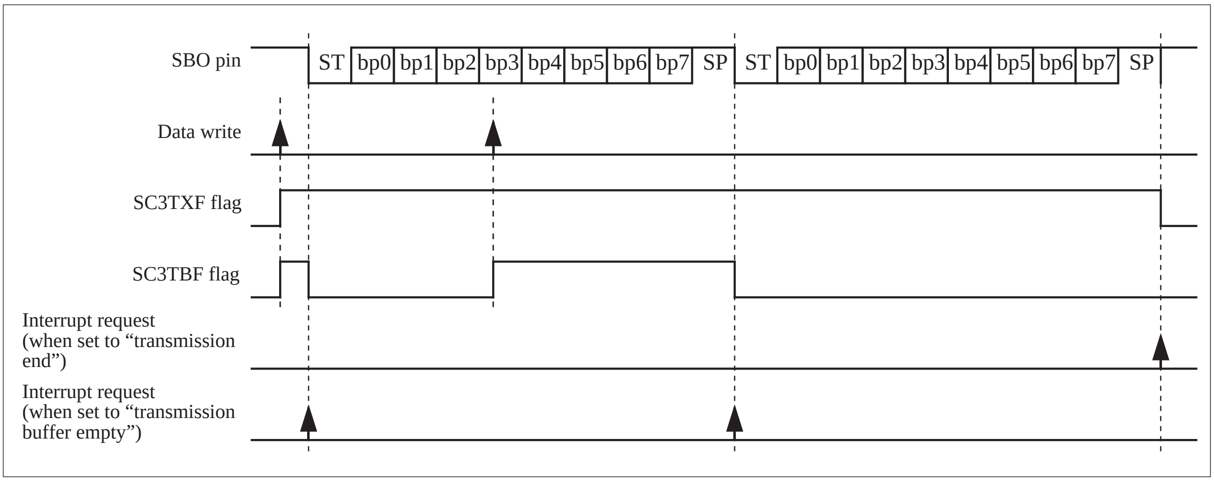

SBO pin
ST
bp0
bp1
bp2
bp3
bp4
bp5
bp6
bp7
SP
ST
bp0
bp1
bp2
bp3
bp4
bp5
bp6
bp7
SP
Data write
SC3TXF flag
SC3TBF flag
Interrupt request (when set to “transmission end”)
Interrupt request (when set to “transmission buffer empty”)

Fig. 13-4-4 Timing Chart (19)

When transmission is enabled, transmission starts when data is written to SC3TXB.
Continuous transmission is possible by writing data to SC3TXB again while transmission is in progress.
During a 7-bit transfer, the MSB (bit 7) is ignored.

The SC3TXF flag is set to "1" when data is written to SC3TXB, and is set to "0" at the end of transmission.
The SC3TBF flag is set to "1" when data is written to SC3TXB, and is set to "0" at the start of transmission.

<Reception>

- Transfer with 7-bit data length, parity on, and 2 stop bit

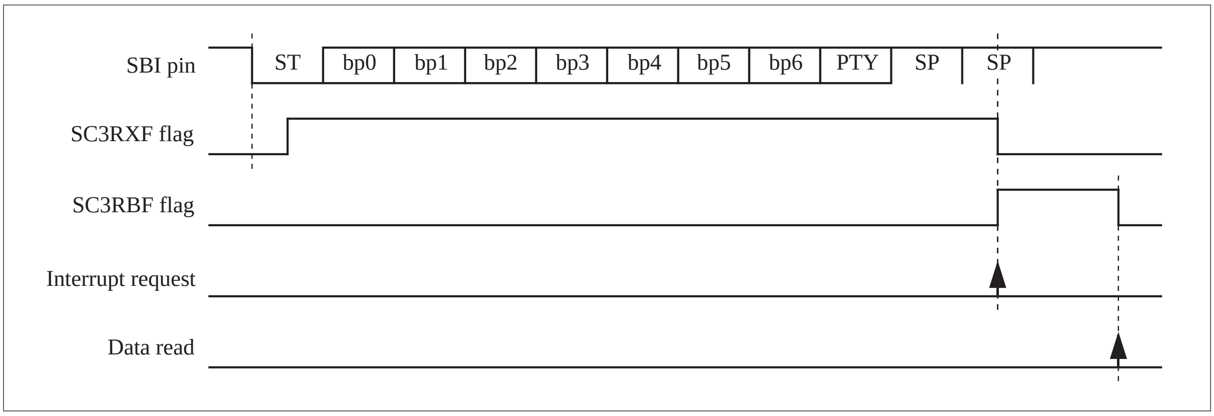


Fig. 13-4-5 Timing Chart (20)

- Two-byte transfer with 8-bit data length, parity off, and 1 stop bit

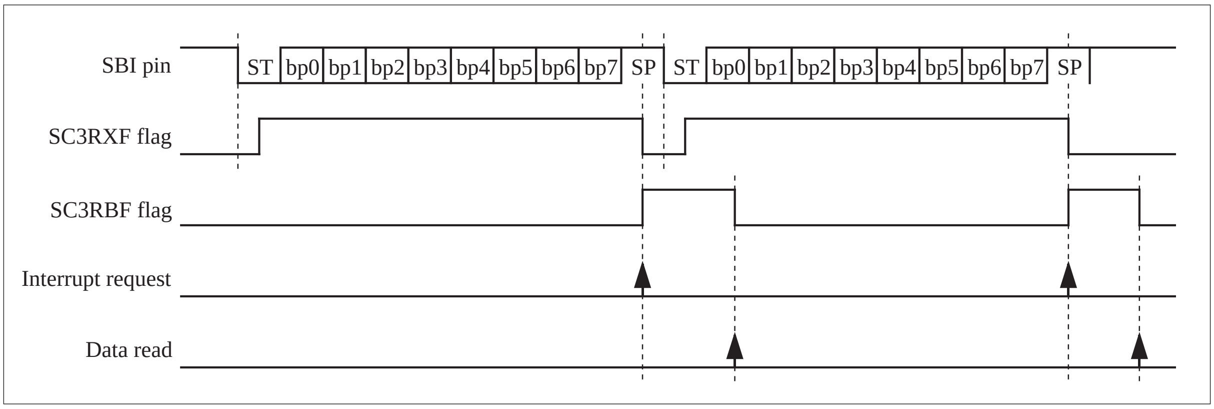


Fig. 13-4-6 Timing Chart (21)

After reception end (when the SC3RBF flag is "1"), the received data is fetched by reading the SC3RXB. In the case of a 7-bit transfer, the MSB (bit 7) is "0".

The SC3RXF flag is set to "1" at the start of reception (when the start bit is detected), and is set to "0" at the end of reception.
The SC3RBF flag is set to "1" at the end of reception, and is set to "0" when SC3RXB is read.

■ When a reception error is generated

- Transfer with 7-bit data length, parity on, and 2 stop bit

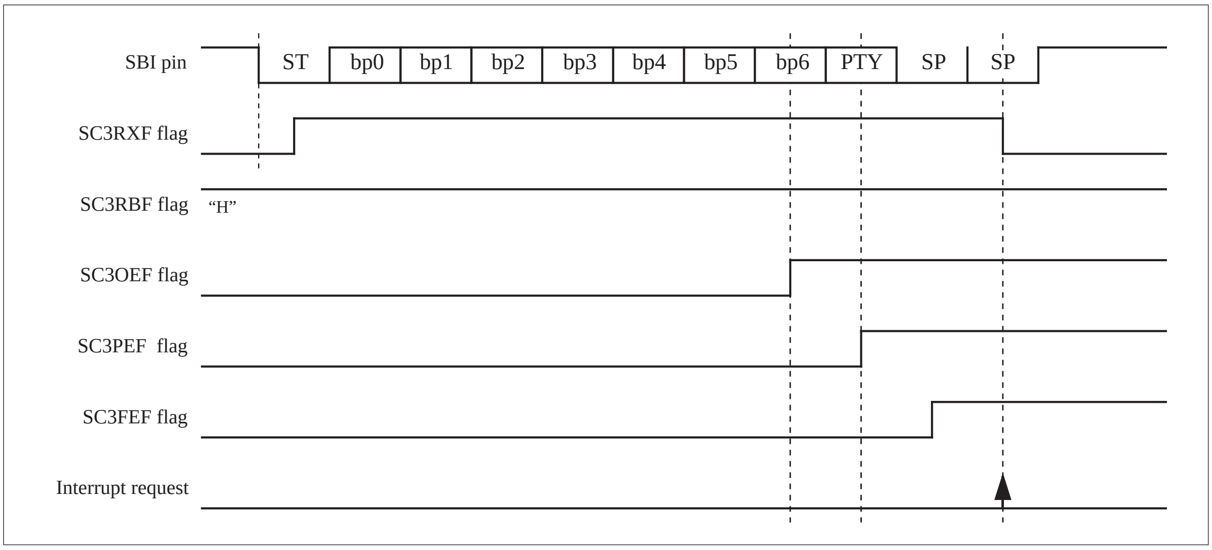


Fig. 13-4-7 Timing Chart (22)

When "reception end" is set as the reception interrupt source, an interrupt request is generated when reception ends, regardless of whether or not an error occurred.

When "reception end with error" is set as the reception interrupt source, an interrupt request is generated when reception ends with an error having occurred. (An interrupt request is not generated at the moment that the error occurred.)

An overrun error is generated when reception of the next data is completed before previously received data is read from the SC3RXB. In this event, the previously received data is lost. The overrun error indicator flag (SC3OEF) is updated at the moment the final data bit is received.

A parity error is generated when 0-fixed parity is set and a "1" is received, when 1-fixed parity is set and a "0" is received, when even parity is set and an odd number of ones is received, or when odd parity is set and an even number of ones is received. The parity error indicator flag (SC3PEF) is updated at the moment the parity bit is received.

A feaming error is generated when "0" was received for the stop bit. The framing error indicator flag (SC3FEF) is updated at the moment the stop bit is received.

■ Transmission interruption function

SC3TWS and SC3TWE in the SC3CTR serial 3 control register can be used to interrupt and resume transmissions according to the status of external pin IRQ7.

A transmission is interrupted by masking the transmission end or transmission buffer empty interrupt.

The specifications of this microcontroller prohibit the writing of data to the transmission buffer when the interrupt signal is masked through external pin control.

14. A/D Converter

14.1 Overview

The A/D converter is a 10-bit charge redistribution-type A/D converter that can process analog signals on a maximum of four channels.

The A/D conversion reference clock can be selected from 1/2, 1/4, 1/8, or 1/16 of IOCLK. When IOCLK = 10 MHz, A/D conversion is performed with a maximum conversion speed of 2.8 μs/ch. (1/2 x IOCLK is selected as the A/D conversion reference clock, and the number of sampling cycles is 2 cycles.)

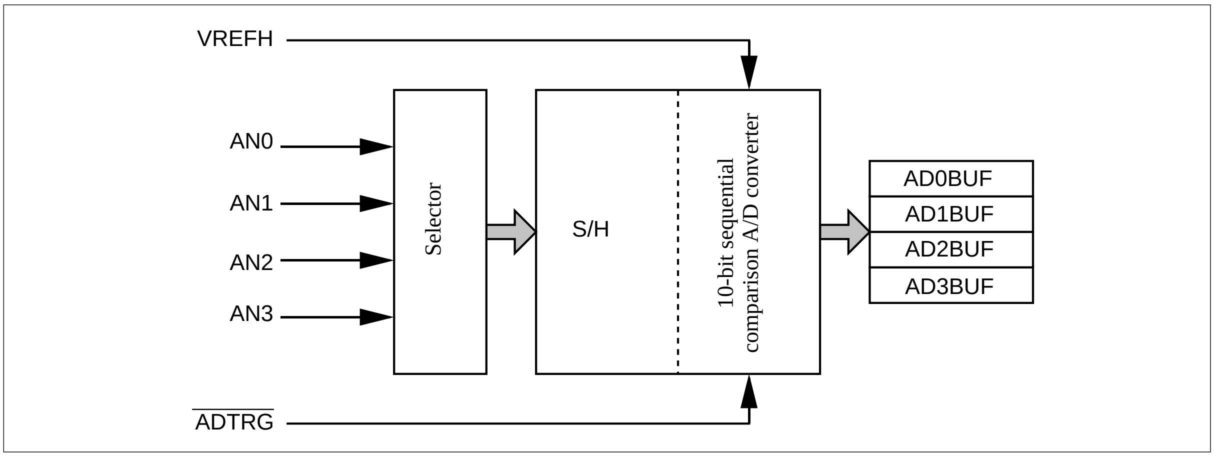

VREFH
AN0
AN1
AN2
AN3
Selector
S/H
10-bit sequential comparison A/D converter
AD0BUF
AD1BUF
AD2BUF
AD3BUF
$\overline{\text{ADTRG}}$

Fig. 14-1-1 A/D Converter Configuration Diagram

14.2 Features

- S/H Built in
- Conversion accuracy

  10 bits ± 5 LSB (Linearity error)

  The value of VREFH divided into 1024 steps is stored in AD0BUF to AD3BUF.
- Conversion reference clock

  Selectable from 1/2, 1/4, 1/8, or 1/16 of IOCLK

  Set this parameter so that one cycle is at least 200 ns.

  (Example: When IOCLK is 15 MHz, set this parameter 1/4 or 1/8 or 1/16.)
- Number of sampling cycles

  Select either two or four conversion reference clock cycles.

  Set this parameter so that the sampling cycle is at least 400 ns.
- Conversion time 2.8 µs/channel

  (When IOCLK is 10 MHz; conversion reference clock is 1/2 of IOCLK and the number of sampling cycles is 2 cycles)

  3.74 µs/channel

  (When IOCLK is 15 MHz; conversion reference clock is 1/4 of IOCLK and the number of sampling cycles is 2 cycles)
- Operating modes 14 modes

  Channel 0 one-time conversion, Channel 0 continuous conversion
  Channel 1 one-time conversion, Channel 1 continuous conversion,
  Channel 0 to 1 one-time conversion, Channel 0 to 1 continuous conversion
  Channel 2 one-time conversion, Channel 2 continuous conversion,
  Channel 0 to 2 one-time conversion, Channel 0 to 2 continuous conversion
  Channel 3 one-time conversion, Channel 3 continuous conversion,
  Channel 0 to 3 one-time conversion, Channel 0 to 3 continuous conversion
- Conversion start 1) Timer 2 underflow

  2) Trigger input (falling edge) to external pin ($\overline{\text{ADTRG}}$ pin)

  3) Register setting by instruction
- Interrupts An interrupt request is generated when a conversion is completed on one channel or on one series of channels.

14.3 Block Diagram

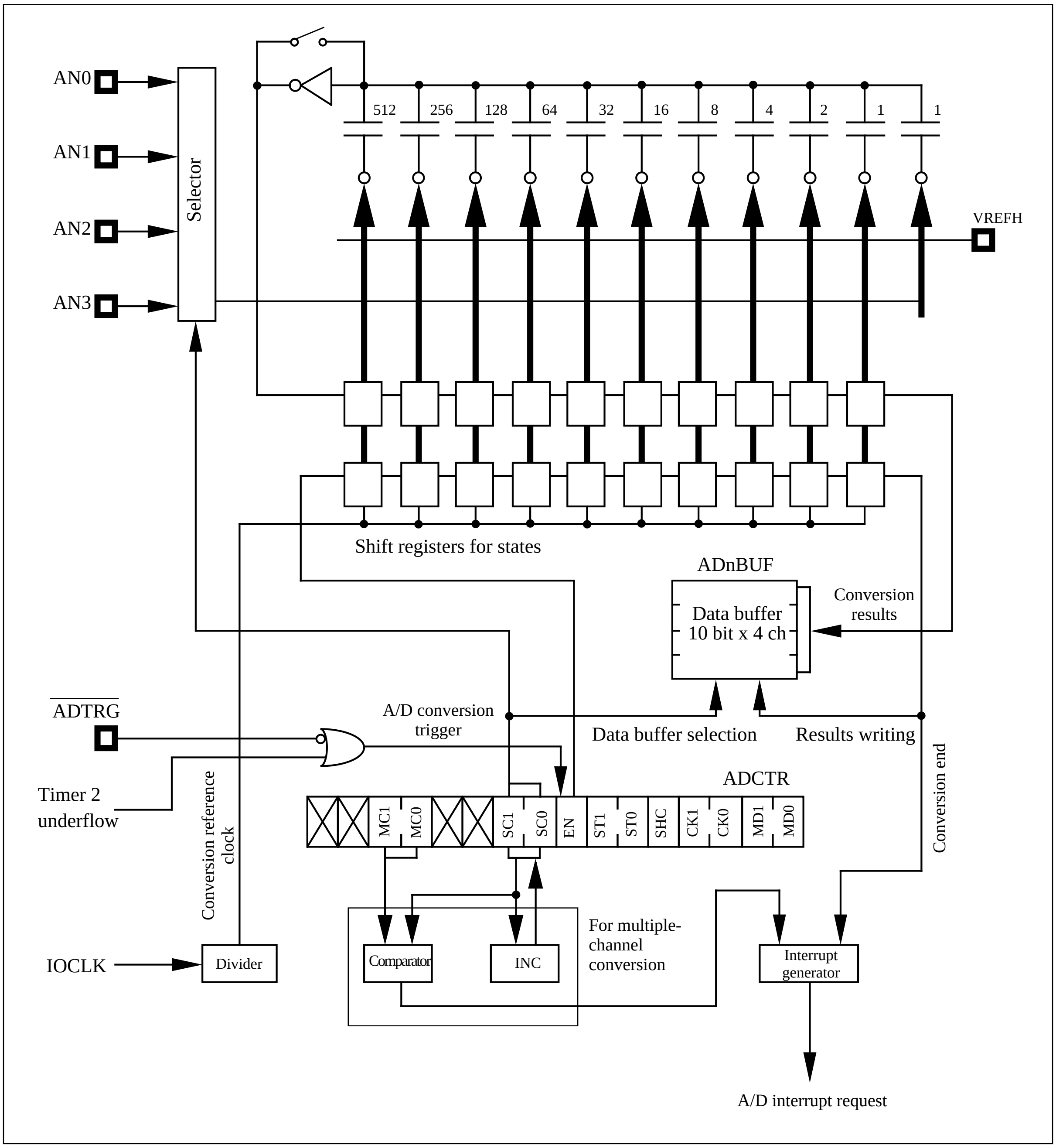


Fig. 14-3-1 The Block Diagram of A/D Converter

14.4 Description of Registers

Table 14-4-1 lists the registers for this A/D converter.

Table 14-4-1 A/D Register List

Address	Name	Symbol	Number of bits	Initial value	Access size
x'34000400	A/D conversion control register	ADCTR	16	x'0000	8, 16
x'34000410	A/D0 conversion data buffer	AD0BUF	16	x'0000	8, 16
x'34000414	A/D1 conversion data buffer	AD1BUF	16	x'0000	8, 16
x'34000418	A/D2 conversion data buffer	AD2BUF	16	x'0000	8, 16
x'3400041C	A/D3 conversion data buffer	AD3BUF	16	x'0000	8, 16

A/D conversion control register

Register symbol: ADCTR
Address: x'34000400
Purpose: This register sets the A/D conversion operation control conditions.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	–	–	AD MC1	AD MC0	–	–	AD SC1	AD SC0	AD EN	AD ST1	AD ST0	AD SHC	AD CK1	AD CK0	AD MD1	AD MD0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit No. Bit name Description

0 ADMD0 Operating mode selection (LSB)

1 ADMD1 Operating mode selection (MSB)

00: Any one channel/one-time conversion
01: Multiple channels/one-time conversion
10: Any one channel/continuous conversion
11: Multiple channels/continuous conversion

2 ADCK0 Conversion reference clock selection (LSB)

3 ADCK1 Conversion reference clock selection (MSB)

00: IOCLK/2
01: IOCLK/4
10: IOCLK/8
11: IOCLK/16

4 ADSHC Sampling cycle number selection

0: Two conversion reference clock cycles
1: Four conversion reference clock cycles

5 ADST0 Conversion start trigger selection (LSB)

6 ADST1 Conversion start trigger selection (MSB)

00: Conversion start by software
01: External trigger (ADEN flag is set by input of falling edge to $\overline{\text{ADTRG}}$ pin)
10: Timer trigger (ADEN flag is set by timer 2 underflow)
11: Setting prohibited

7	ADEN	Conversion start/execution flag (conversion can be started by writing a "1" to this flag) 0: Conversion stopped 1: Conversion start/in progress
8	ADSC0	Selection of conversion channel when converting any one channel/ indicator of current conversion channel when converting multiple channels (LSB)
9	ADSC1	Selection of conversion channel when converting any one channel/ indicator of current conversion channel when converting multiple channels (MSB) 00: AN0 01: AN1 10: AN2 11: AN3
10	—	Must be set to "0".
11	—	Must be set to "0".
12	ADMC0	Conversion channels when converting multiple channels (LSB)
13	ADMC1	Conversion channels when converting multiple channels (MSB) 00: AN0 01: AN0 to AN1 10: AN0 to AN2 11: AN0 to AN3
14	—	Must be set to "0".
15	—	Must be set to "0".

Note: When a multiple number of channels are to be converted, set "00" initially for ADSC1 to ADSC0.

A/Dn conversion data buffer (n = 0, 1, 2, 3)

Register symbol: ADnBUF
Address: x'34000410 (n = 0), x'34000414 (n = 1), x'34000418 (n = 2), x'3400041C (n = 3)
Purpose: This register stores the A/D conversion result for the ANn pin (channel n).

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	ADn BUF15	ADn BUF14	ADn BUF13	ADn BUF12	ADn BUF11	ADn BUF10	ADn BUF9	ADn BUF8	ADn BUF7	ADn BUF6	–	–	–	–	–	–
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

The A/D conversion result (10-bit data) is stored in bits 15 to 6.
If bits 5 to 0 are read, zeroes are returned.

14.5 Description of Operation

■ Operating mode selection

(1) Any one channel/one-time conversion

If "any one channel/one-time conversion" is selected as the operating mode (ADMD1 to 0), one AN input is converted one time only. Set the conversion channel in the conversion channel selection bits (ADSC1 to 0). (ADMC1 to 0 are ignored.) An A/D interrupt request is generated simultaneously with the completion of conversion.

When starting up conversion through software, set the conversion start trigger selection bits (ADST1 to 0) to "00", and set the conversion start/execution flag (ADEN) to "1".

If the conversion start trigger selection bits (ADST1 to 0) are set to "external trigger", then the conversion start/execution flag (ADEN) is set to "1" when a falling edge is input to the ADTRG pin, A/D conversion then starts.

If the conversion start trigger selection bits (ADST1 to 0) are set to "timer trigger," then the conversion start/ execution flag (ADEN) is set to "1" when a timer 2 underflow occurs, A/D conversion then starts.

The conversion start/execution flag (ADEN) is "1" while conversion is in progress, and is then set to "0" after conversion is completed.

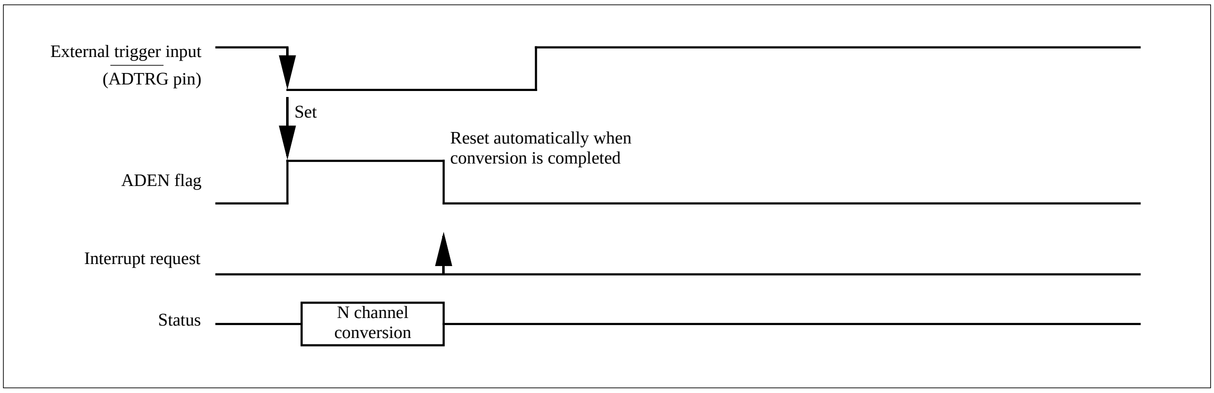


Fig. 14-5-1 External Trigger Input Conversion Example

(2) Multiple channels/one-time conversion for each channel

If "multiple channels/one-time conversion for each channel" is selected as the operating mode (ADMD1 to 0), a number of AN inputs, starting from AN0, are converted one time only. Set channel 0 in the conversion channel selection bits used for converting any one channel (ADSC1 to 0), and set the number of channels to be converted in the conversion channel selection bits (ADMC1 to 0). (Conversion starts with channel 0.) An A/D interrupt request is generated simultaneously with the completion of conversion of all channels.

When starting up conversion through software, set the conversion start trigger selection bits (ADST1 to 0) to "00", and set the conversion start/execution flag (ADEN) to "1".

If the conversion start trigger selection bits (ADST1 to 0) are set to "external trigger," then the conversion start/execution flag (ADEN) is set to "1" when a falling edge is input to the $\overline{\text{ADTRG}}$ pin, A/D conversion then starts.

And if the conversion start trigger selection bits (ADST1 to 0) are set to "timer trigger," then the conversion start/execution flag (ADEN) is set to "1" when a timer 2 underflow occurs, A/D conversion then starts.

The conversion start/execution flag (ADEN) is "1" while conversion is in progress, and is then set to "0" after conversion of all channels is completed. The conversion channel selection bits that are used for selecting any one channel for conversion (ADSC1 to 0) indicate the current channel number being converted, and are then set to "00" when conversion of all channels is completed.

Note: If multiple channels are to be converted, be certain to include a capacitor of at least 0.1 μF between each AN pin and AVSS. Each capacitor should be placed as closely as possible to each AN pin.

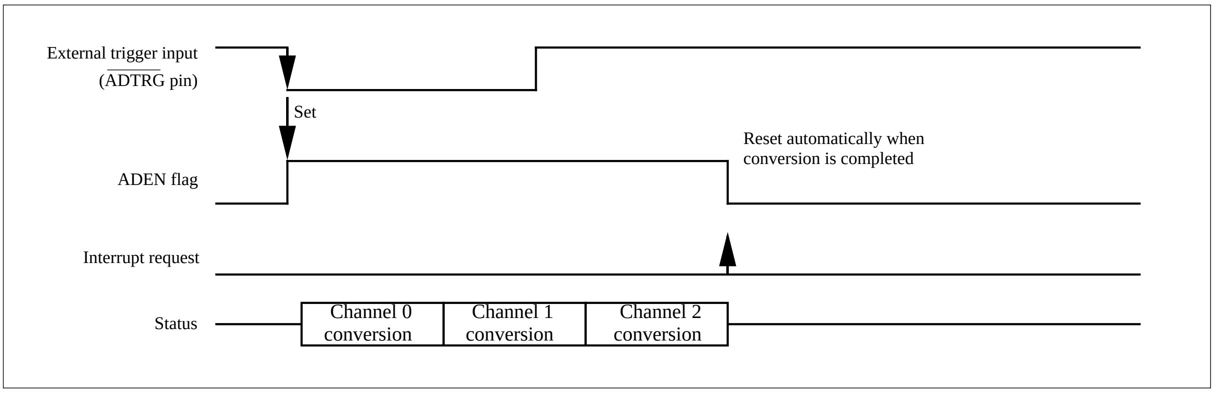


Fig. 14-5-2 External Trigger Input Conversion Example (for Channels 0 to 2, One Time Each)

(3) Any one channel/continuous conversion

If "any one channel/continuous conversion" is selected as the operating mode (ADMD1 to 0), one AN input is converted continuously. Set the conversion channel in the conversion channel selection bits (ADSC1 to 0). (ADMC1 to 0 are ignored.) An A/D interrupt request is generated each time conversion is completed.

When starting up conversion through software, set the conversion start trigger selection bits (ADST1 to 0) to "00", and set the conversion start/execution flag (ADEN) to "1".

If the conversion start trigger selection bits (ADST1 to 0) are set to "external trigger," then the conversion start/execution flag (ADEN) is set to "1" when a falling edge is input to the $\overline{\text{ADTRG}}$ pin, A/D conversion then starts.

And if the conversion start trigger selection bits (ADST1 to 0) are set to "timer trigger," then the conversion start/execution flag (ADEN) is set to "1" when a timer 2 underflow occurs, A/D conversion then starts.

The conversion start/execution flag (ADEN) is "1" while conversion is in progress, and is not cleared by hardware. Therefore, set the conversion start/execution flag (ADEN) to "0" when stopping the conversion operation.

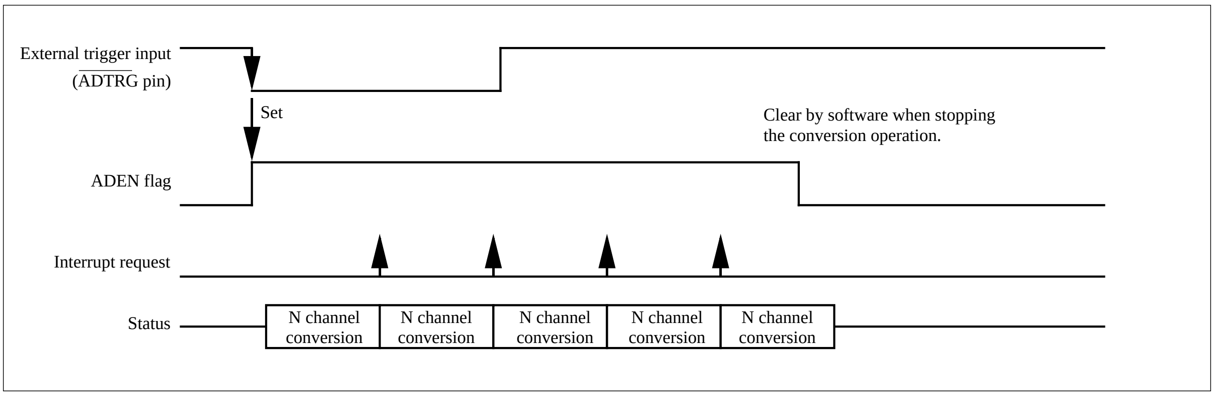


Fig. 14-5-3 External Trigger Input Conversion Example

(4) Multiple channels/continuous conversion

If "multiple channels/continuous conversion" is selected as the operating mode (ADMD1 to 0), a number of AN inputs, starting from AN0, are converted continuously. Set channel 0 in the conversion channel selection bits used for converting any one channel (ADSC1 to 0), and set the number of channels to be converted in the conversion channel selection bits (ADMC1 to 0). (Conversion starts with channel 0.) An A/D interrupt request is generated each time one round of conversion of all channels is completed.

When starting up conversion through software, set the conversion start trigger selection bits (ADST1 to 0) to "00", and set the conversion start/execution flag (ADEN) to "1".

If the conversion start trigger selection bits (ADST1 to 0) are set to "external trigger," then the conversion start/execution flag (ADEN) is set to "1" when a falling edge is input to the $\overline{\text{ADTRG}}$ pin, A/D conversion then starts.

And if the conversion start trigger selection bits (ADST1 to 0) are set to "timer trigger," then the conversion start/execution flag (ADEN) is set to "1" when a timer 2 underflow occurs, A/D conversion then starts.

The conversion start/execution flag (ADEN) is "1" while conversion is in progress, and is not cleared by hardware. Therefore, set the conversion start/execution flag (ADEN) to "0" when stopping the conversion operation. The conversion channel selection bits that are used for selecting any one channel for conversion (ADSC1 to 0) indicate the current channel number being converted, and are then set to "00" when conversion of all channels is completed.

Note: If multiple channels are to be converted, be certain to include a capacitor of at least 0.1 μF between each AN pin and AVSS. Each capacitor should be placed as closely as possible to each AN pin.

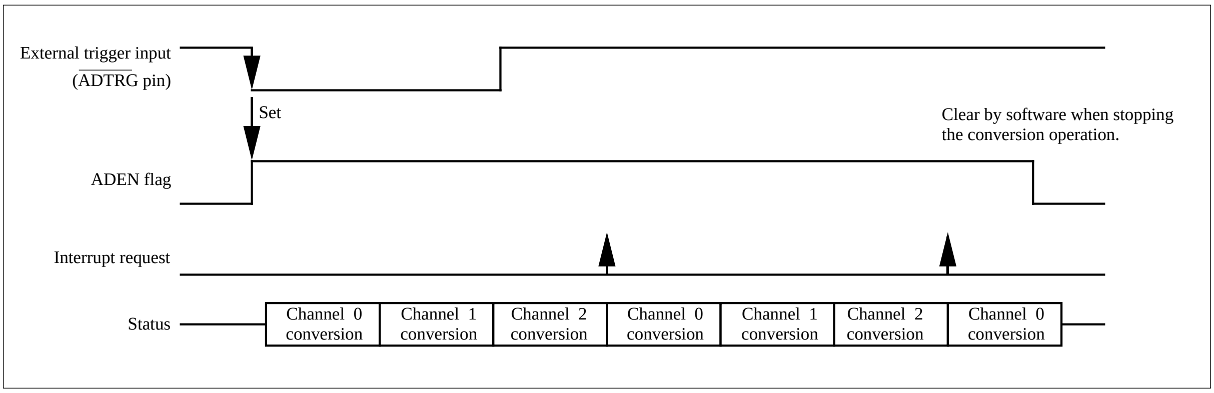


Fig. 14-5-4 External Trigger Input Conversion Example (for Channels 0 to 2, Continuous Conversion)

■ Conversion reference clock selection, sampling cycle number selection

The A/D conversion time is [(12 + number of sampling cycles) x IOCLK/clock selection]/channel. For example, if the conversion reference clock is set as 1/8 of IOCLK and the number of sampling cycles is set as two cycles, the A/D conversion time is IOCLK x 112 cycles/channel.

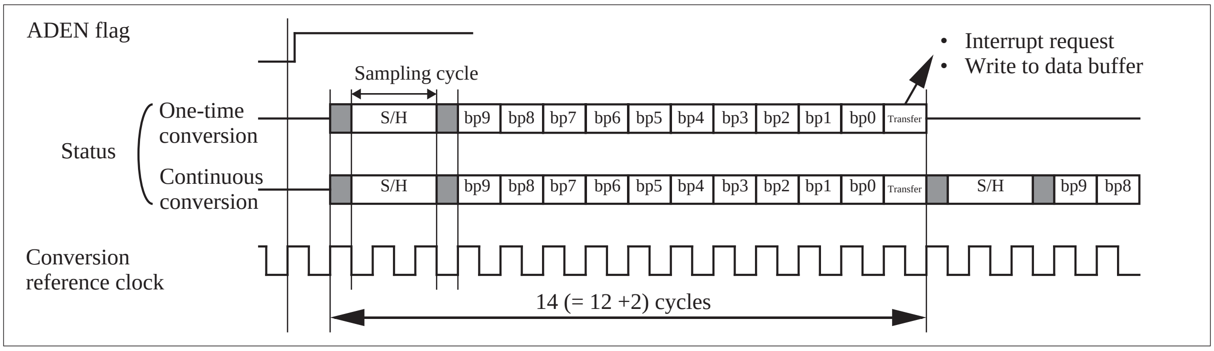


Fig. 14-5-5 Conversion Timing When Using Two Sampling Cycles

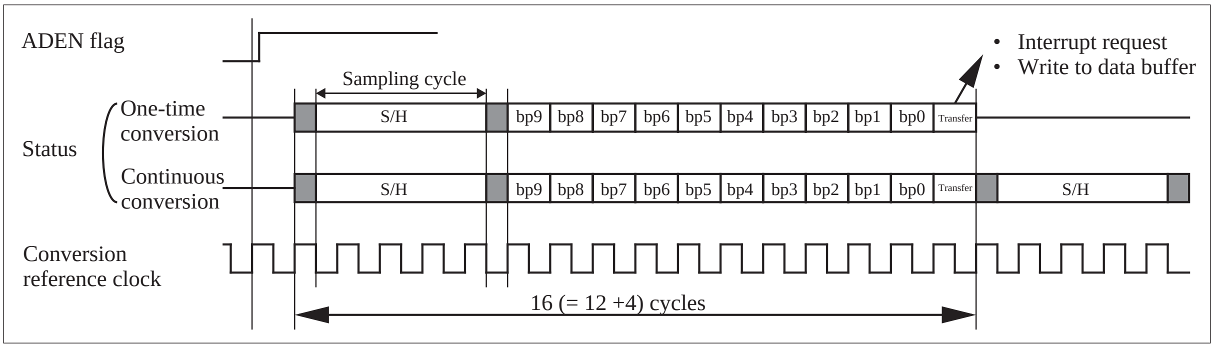


Fig. 14-5-6 Conversion Timing When Using Four Sampling Cycles

Set the conversion reference clock so that one cycle is at least 200 ns.

Set the number of sampling cycles so that one sampling cycle is at least 400 ns (when the output impedance of the external device that drives the AN pin is 1 kΩ or less).

If the output impedance of the external device that drives the AN pin is greater than 1 kΩ, it is necessary to lengthen the sampling cycle.

When A/D conversion (ADEN = "1") is started up from the stopped state (ADEN = "0"), a wait state of a maximum of one conversion reference clock cycle is inserted between the point when ADEN goes to "1" and the actual start of conversion.

[Notes]

If a falling edge is input to the $\overline{\text{ADTRG}}$ pin before the conversion start trigger selection (ADST1 to 0) is switched to "external trigger" ("01"), the ADEN flag is set at the same time that the switch is made, and A/D conversion starts.

Fig. 14-5-7 shows an example of a single conversion. In this case, the ADEN conversion start/execution flag is set at the same time that ADST1 to 0 are switched, and is reset when A/D conversion is completed.

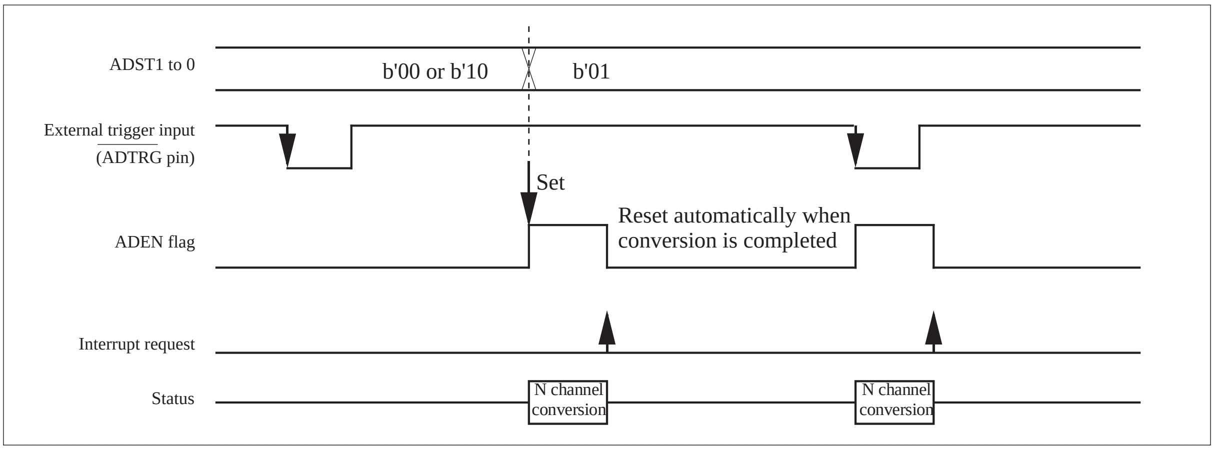


Fig. 14-5-7   Example of Conversion by Switching to External Trigger Mode (Single Conversion)

Fig. 14-5-8 shows an example of continuous conversion. The ADEN flag is set at the same time that ADST1 to 0 are switched, and then A/D conversion starts. In the case of continuous conversion, conversion is stopped by writing "0" to the ADEN flag.

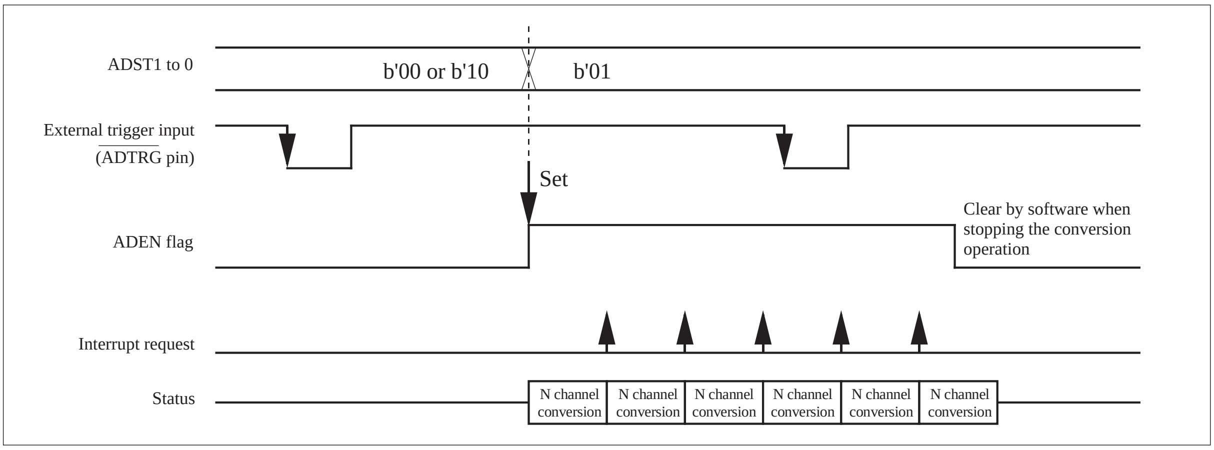


Fig. 14-5-8   Example of Conversion by Switching to External Trigger Mode (Continuous Conversion)

15. I/O Ports

15.1 Overview

The MN103001G and MN1030F01K have a total of 13 internal I/O ports: 0 to 9, A, B and C. These ports can all be accessed by programs as internal I/O memory space.

Port 0 is a 3-bit general-purpose output port; ports 1, 2, A, and B are 8-bit general-purpose input/output ports; port 3 is a 1-bit general-purpose input/output port; ports 4 and 5 are 6-bit general-purpose input/output ports; port 6 is a 4-bit general-purpose input/output port; ports 7 and C are 4-bit general-purpose output ports; port 8 is a 4-bit general-purpose input port; and port 9 consists of 8 bits: P97 and P94 to P92 are general-purpose output ports, while P96, P95, P91, and P90 are general-purpose input/output ports.

Each port pin has additional functions, described below. The function of these pins can be switched via the control register within the I/O ports.

Port 0 (P0)

This port is also used for address bus signals A[22:20] and for the DRAM CAS signal $\overline{\text{CAS}}$.

Port 1 (P1)

This port is also used for data bus signals D[7:0], the address strobe signal AS, and read/write select RWSEL.

Port 2 (P2)

This port is also used for data bus signals D[15:8].

Port 3 (P3)

This port is also used for the bus grant signal $\overline{\text{BG}}$.

Port 4 (P4)

This port is also used for the serial interface input/output signals SBI1, SBO1, SBT1, SBI0, SBO0, SBT0; the DRAM CAS signals (for 2$\overline{\text{CAS}}$) $\overline{\text{DCAS1}}$ and $\overline{\text{DCAS0}}$; and the DRAM write signal (for 2$\overline{\text{CAS}}$) $\overline{\text{DWE}}$.

Port 5 (P5)

This port is also used for the serial interface input/output signals SBI3, SBO3, SBT3, SBI2, SBO2, SBT2; and the timer input/output signals TM13IO, TM12IO, TM11IO, TM5IO, TM4IO, TM3IO, TM2IO, TM1IO, and TM0IO.

Port 6 (P6)

This port is also used for external interrupt inputs IRQ3 to IRQ0; the timer input/output signals TM6IO, TM7IO, TM10IOA, TM10IOB; and the A/D conversion trigger input $\overline{\text{ADTRG}}$.

Port 7 (P7)

This port is also used for address bus signal A23; DRAM RAS signals $\overline{\text{RAS2}}$ and $\overline{\text{RAS1}}$; and chip select signals $\overline{\text{CS3}}$ to $\overline{\text{CS0}}$.

Port 8 (P8)

This port is also used for analog signal inputs AN3 to AN0 and external interrupt inputs IRQ7 to I RQ4.

Port 9 (P9)

This port is also used for extension mode setting signals EXMOD1 and EXMOD0; memory write signals $\overline{\text{WE}}$1 and $\overline{\text{WE0}}$; memory read signal $\overline{\text{RE}}$; bus authority request signal $\overline{\text{BR}}$; data acknowledge signal $\overline{\text{DK}}$; and system clock SYSCLK.

Port A (PA)

This port is also used for address bus signals A[7:0], and address/data signals ADM[7:0].

Port B (PB)

This port is also used for address bus signals A[15:8], and address/data signals ADM[15:8].

Port C (PC)

This port is also used for address bus signals A[19:16].

The I/O ports are provided with the registers listed in Table 15-1-1.

Table 15-1-1 List of Registers (1/2)

Address	Name	Symbol	Number of bits	Initial value	Access size
x'36008000	Port 0 output register	P0OUT	8	x'00	8, 16
x'36008001	Port 1 output register	P1OUT	8	x'00	8
x'36008004	Port 2 output register	P2OUT	8	x'00	8, 16
x'36008005	Port 3 output register	P3OUT	8	x'00	8
x'36008008	Port 4 output register	P4OUT	8	x'00	8, 16
x'36008009	Port 5 output register	P5OUT	8	x'00	8
x'3600800C	Port 6 output register	P6OUT	8	x'00	8, 16
x'3600800D	Port 7 output register	P7OUT	8	x'00	8
x'36008011	Port 9 output register	P9OUT	8	x'00	8
x'36008014	Port A output register	PAOUT	8	x'00	8, 16
x'36008015	Port B output register	PBOUT	8	x'00	8
x'36008018	Port C output register	PCOUT	8	x'00	8
x'36008020	Port 0 output mode register	P0MD	8	x'00	8, 16
x'36008021	Port 1 output mode register	P1MD	8	x'02(x'01)	8
x'36008024	Port 2 output mode register	P2MD	8	x'02(x'01)	8, 16
x'36008025	Port 3 output mode register	P3MD	8	x'01	8
x'36008028	Port 4 output mode register	P4MD	8	x'3F	8, 16
x'36008029	Port 5 output mode register	P5MD	8	x'3F	8
x'3600802C	Port 6 output mode register	P6MD	8	x'0F	8, 16
x'3600802D	Port 7 output mode register	P7MD	8	x'00	8
x'36008030	Port 8 analog/digital input control register	P8AD	8	x'0F	8, 16
x'36008031	Port 9 output mode register	P9MD	8	x'60	8
x'36008034	Port A output mode register	PAMD	8	x'00(x'02)	8, 16
x'36008035	Port B output mode register	PBMD	8	x'00(x'02)	8
x'36008038	Port C output mode register	PCMD	8	x'00	8
x'36008040	Port 0 dedicated output control register	P0SS	8	x'00	8
x'36008048	Port 4 dedicated output control register	P4SS	8	x'00	8, 16
x'36008049	Port 5 dedicated output control register	P5SS	8	x'3F	8
x'3600804D	Port 7 dedicated output control register	P7SS	8	x'00	8
x'36008061	Port 1 input/output control register	P1DIR	8	x'00	8
x'36008064	Port 2 input/output control register	P2DIR	8	x'00	8, 16
x'36008065	Port 3 input/output control register	P3DIR	8	x'00	8
x'36008068	Port 4 input/output control register	P4DIR	8	x'00	8, 16
x'36008069	Port 5 input/output control register	P5DIR	8	x'00	8
x'3600806C	Port 6 input/output control register	P6DIR	8	x'00	8
x'36008071	Port 9 input/output control register	P9DIR	8	x'9C	8
x'36008074	Port A input/output control register	PADIR	8	x'00	8, 16
x'36008075	Port B input/output control register	PBDIR	8	x'00	8

The values in parentheses apply when address/data multiplex mode.

Table 15-1-1 List of Registers (2/2)

Address	Name	Symbol	Number of bits	Initial value	Access size
x'36008081	Port 1 pin register	P1IN	8	x'XX	8
x'36008084	Port 2 pin register	P2IN	8	x'XX	8, 16
x'36008085	Port 3 pin register	P3IN	8	x'0X	8
x'36008088	Port 4 pin register	P4IN	8	x'XX	8, 16
x'36008089	Port 5 pin register	P5IN	8	x'XX	8
x'3600808C	Port 6 pin register	P6IN	8	x'0X	8
x'36008090	Port 8 pin register	P8IN	8	x'0X	8, 16
x'36008091	Port 9 pin register	P9IN	8	x'XX	8
x'36008094	Port A pin register	PAIN	8	x'XX	8, 16
x'36008095	Port B pin register	PBIN	8	x'XX	8

15.2 Port 0

15.2.1 Block Diagram

Fig. 15-2-1 and Fig 15-2-2 show block diagrams for port 0.

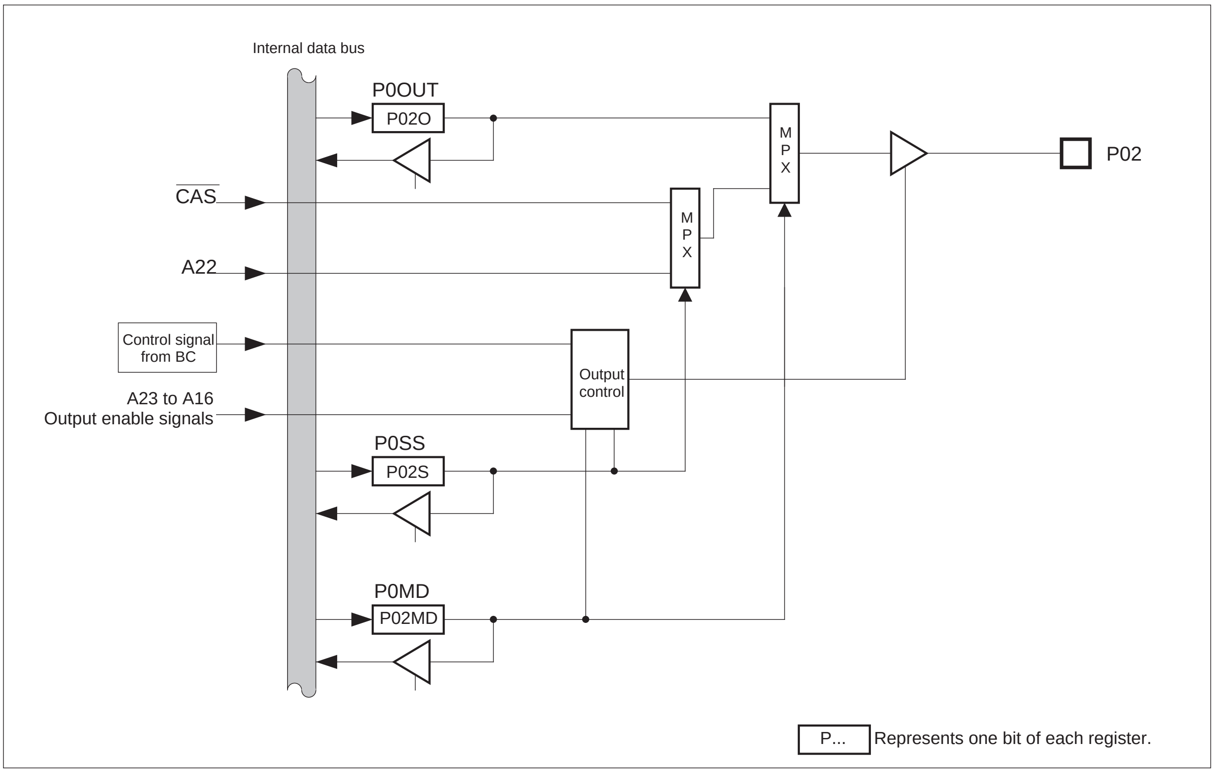


Fig. 15-2-1 Port 0 Block Diagram (P02)

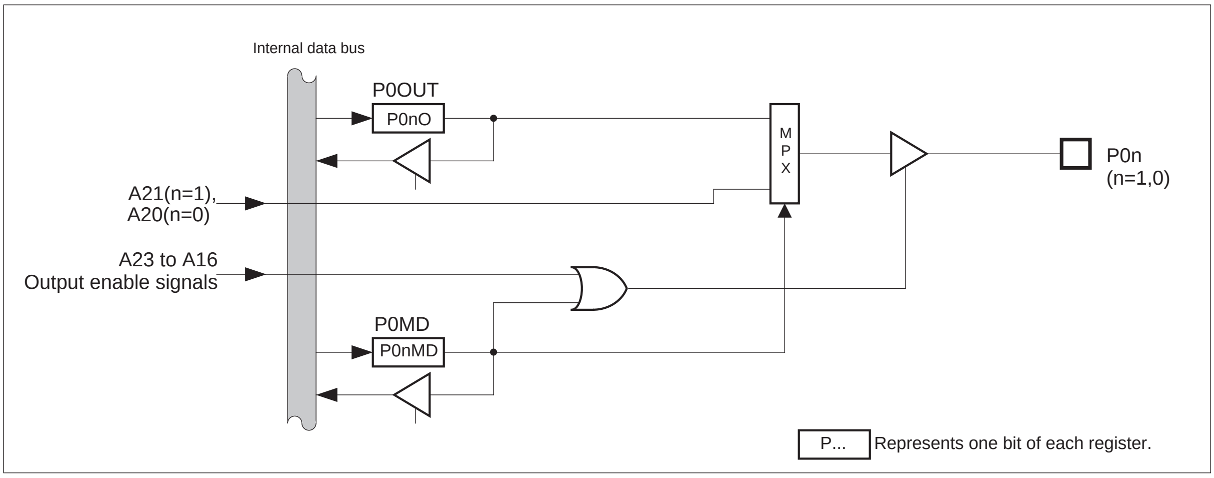


Fig. 15-2-2 Port 0 Block Diagram (P01, P00)

15.2.2 Register Descriptions

Port 0 is a general-purpose output port that is also used for address bus A [22:20], DRAM CAS signal $\overline{\text{CAS}}$.

Each register for port 0 is described below.

Port 0 output register

Register symbol: P0OUT
Address: x'36008000
Purpose: This register sets the data to be output on port 0.

Bit No.	7	6	5	4	3	2	1	0
Bit name	-	-	-	-	-	P02O	P01O	P00O
Reset	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R/W	R/W	R/W

Port 0 output mode register

Register symbol: P0MD
Address: x'36008020
Purpose: This register selects the content output on the port 0 pins with P0SS.

Bit No.	7	6	5	4	3	2	1	0
Bit name	-	-	-	-	-	P02MD	P01MD	P00MD
Reset	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R/W	R/W	R/W

Port 0 dedicated output control register

Register symbol: P0SS
Address: x'36008040
Purpose: This register selects the content output on the port 0 pins.
Valid when the P0nMD bit is "0".

Bit No.	7	6	5	4	3	2	1	0
Bit name	-	-	-	-	-	P02S	-	-
Reset	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R/W	R	R

P02MD; P02S 00: Address output (A22)
01: DRAM CAS signal output [for 2$\overline{\mathrm{WE}}$] ($\overline{\mathrm{CAS}}$)
1x: General-purpose output port (P02)

P01MD 0: Address output (A21)
1: General-purpose output port (P01)

P00MD 0: Address output (A20)
1: General-purpose output port (P00)

15.2.3 Pin Configurations

Table 15-2-1 shows the pin configurations for port 0.

Table 15-2-1 Port 0 Configuration

Port	Pin No.	P0n	P0nMD = "1"	P0nMD = "0"			
				P02S="1"		P02S="0"	
Port 0	100	P00	General-purpose output port	A20	Address output		
	99	P01	General-purpose output port	A21	Address output		
	97	P02	General-purpose output port	$\overline{CAS}$	DRAM CAS signal output (for 2$\overline{WE}$)	A22	Address output

[Note 1]

▨ : When reset (whether in address/data separate mode or address/data multiplex mode)

[Note 2]

When the bus authority is granted, $\overline{CAS}$, and A22 to A20 go to high impedance.

15.3 Port 1

15.3.1 Block Diagram

Figs. 15-3-1 and 15-3-2 show block diagrams for port 1.

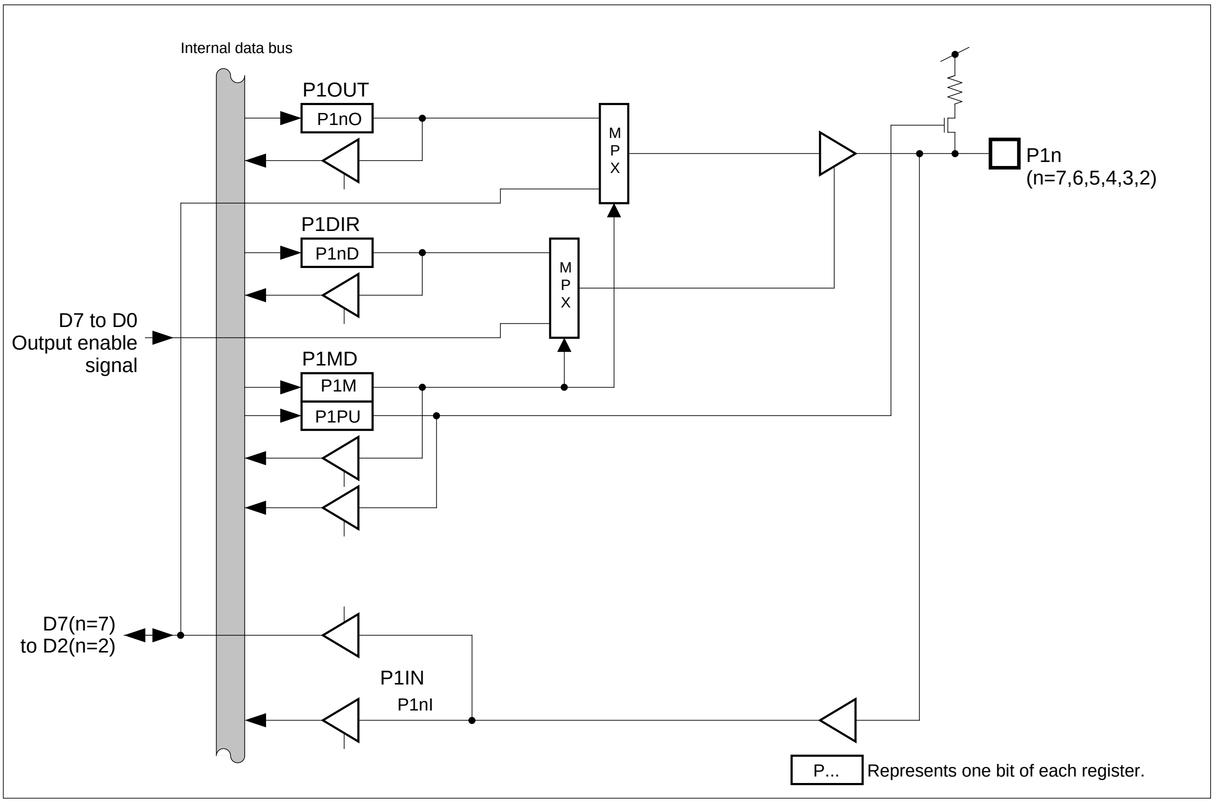


Fig. 15-3-1 Port 1 Block Diagram (P17 to P12)

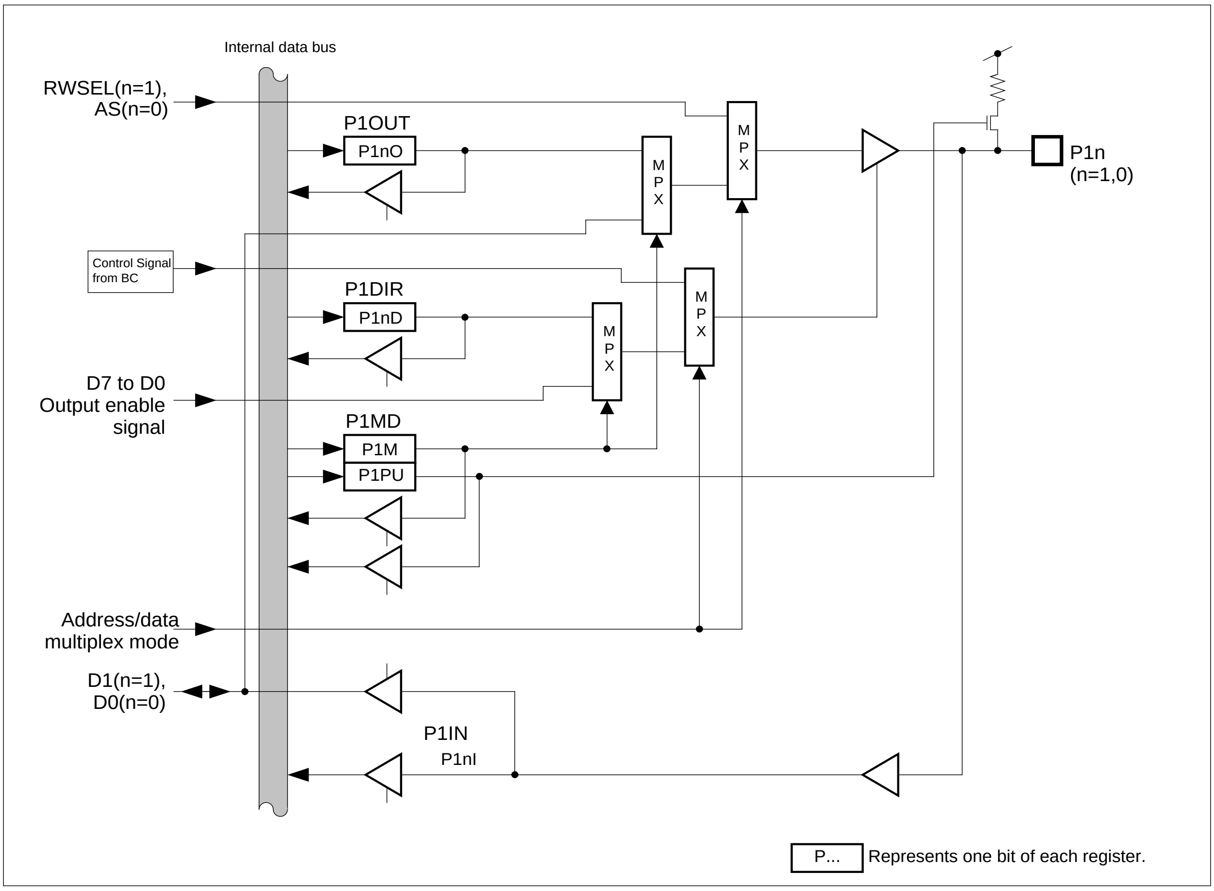


Fig. 15-3-2 Port 1 Block Diagram (P11, and P10)

15.3.2 Register Descriptions

Port 1 is a general-purpose input/output port that is also used for data bus signals D[7:0], address strobe signal AS, and read/write select RWSEL.

Each register for port 1 is described below.

Port 1 output register

Register symbol: P1OUT
Address: x'36008001
Purpose: This register sets the data to be output on port 1.

Bit No.	7	6	5	4	3	2	1	0
Bit name	P17O	P16O	P15O	P14O	P13O	P12O	P11O	P10O
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Port 1 pin register

Register symbol: P1IN
Address: x'36008081
Purpose: This register reads the values of the port 1 pins.

Bit No.	7	6	5	4	3	2	1	0
Bit name	P17I	P16I	P15I	P14I	P13I	P12I	P11I	P10I
Reset	X	X	X	X	X	X	X	X
Access	R	R	R	R	R	R	R	R

Port 1 input/output control register

Register symbol: P1DIR

Address: x'36008061

Purpose: This register sets the port 1 pins for input or output.
(0:input; 1: output)

Bit No.	7	6	5	4	3	2	1	0
Bit name	P17D	P16D	P15D	P14D	P13D	P12D	P11D	P10D
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Port 1 output mode register

Register symbol: P1MD

Address: x'36008021

Purpose: When P1M is "1", the port 1 pins are set as a general-purpose port; when P1M is "0", the port 1 pins are set as data pins.
When P1PU is "1", the port 1 pins are pulled up.

Note: The setting of P1PU that pulls up (or does not pull up) the port 1 pins can be made regardless of the value of P1M.

Bit No.	7	6	5	4	3	2	1	0
Bit name	-	-	-	-	-	-	P1PU	P1M
Reset	0	0	0	0	0	0	1(0)	0(1)
Access	R	R	R	R	R	R	R/W	R/W

() are set in address/data multiplex mode.

In address/data multiplex mode, pin Nos. 95 and 96 are dedicated for use as RWSEL and AS, respectively, and cannot be set as ports.

Note that setting P1M to "0" in address/data multiplex mode is prohibited.

15.3.3 Pin Configurations

Table 15-3-1 shows the pin configurations for port 1.

Table 15-3-1 Port 1 Configuration

Port	Pin No.	P1n	P1M = "1"		P1M = "0"	
			P1nD = "1"	P1nD = "0"		
Port 1	96	P10	General-purpose output port	General-purpose input port	D0 *1	Data input/output
		<<AS>>	<<Address strobe output *Setting invalid>>			
	95	P11	General-purpose output port	General-purpose input port	D1 *1	Data input/output
		<<RWSEL>>	<<Read/write select output *Setting invalid>>			
	94	P12	General-purpose output port	General-purpose input port	D2 *1	Data input/output
	93	P13	General-purpose output port	General-purpose input port	D3 *1	Data input/output
	91	P14	General-purpose output port	General-purpose input port	D4 *1	Data input/output
	90	P15	General-purpose output port	General-purpose input port	D5 *1	Data input/output
	89	P16	General-purpose output port	General-purpose input port	D6 *1	Data input/output
	88	P17	General-purpose output port	General-purpose input port	D7 *1	Data input/output

[Note 1]

▨ : When reset (in address/data separate mode)
"General-purpose input port" is selected in address/data multiplex mode. pin No. 95 and 96, however, are set as RWSEL and AS, respectively.

*1 : In the event of a reset in address/data separate mode, the P1PU bit in the P1MD register is set to "1" and the data pins (the 8 bits D[7:0]) are pulled up.

<<>> : These pins are set in address/data multiplex mode.

Setting invalid: In address/data multiplex mode, pin No. 95 is used only for RWSEL, and pin No. 96 is used only for AS; the P1MD and P1DIR settings are invalid.

[Note 2]

Setting P1M to "0" in address/data multiplex mode is prohibited.

[Note 3]

When the bus authority is granted, D7 to D0, AS, and RWSEL go to high impedance.

15.4 Port 2

15.4.1 Block Diagram

Figs. 15-4-1 shows a block diagrams for port 2.

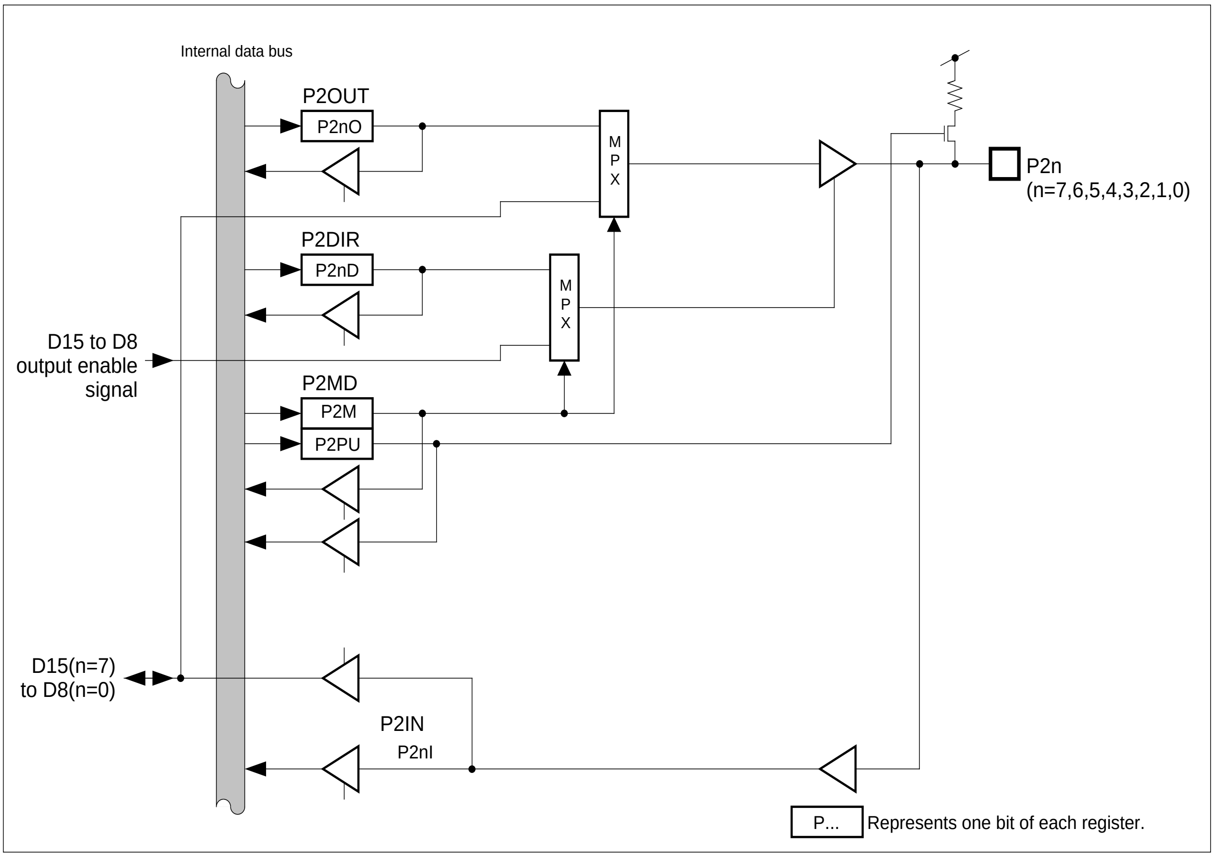


Fig. 15-4-1 Port 2 Block Diagram (P27 to P20)

15.4.2 Register Descriptions

Port 2 is a general-purpose input/output port that is also used for data bus signals D[15:8].

Each register for port 2 is described below.

Port 2 output register

Register symbol: P2OUT
Address: x'36008004
Purpose: This register sets the data to be output on port 2.

Bit No.	7	6	5	4	3	2	1	0
Bit name	P27O	P26O	P25O	P24O	P23O	P22O	P21O	P20O
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Port 2 pin register

Register symbol: P2IN
Address: x'36008084
Purpose: This register reads the values of the port 2 pins.

Bit No.	7	6	5	4	3	2	1	0
Bit name	P27I	P26I	P25I	P24I	P23I	P22I	P21I	P20I
Reset	X	X	X	X	X	X	X	X
Access	R	R	R	R	R	R	R	R

Port 2 input/output control register

Register symbol: P2DIR

Address: x'36008064

Purpose: This register sets the port 2 pins for input or output.
(0: input; 1: output)

Bit No.	7	6	5	4	3	2	1	0
Bit name	P27D	P26D	P25D	P24D	P23D	P22D	P21D	P20D
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Port 2 output mode register

Register symbol: P2MD

Address: x'36008024

Purpose: When P2M is "1", the port 2 pins are set as a general-purpose port; when P2M is "0", the port 2 pins are set as data pins.
When P2PU is "1", the port 2 pins are pulled up.

Note: The setting of P2PU that pulls up (or does not pull up) the port 2 pins can be made regardless of the value of P2M.

Bit No.	7	6	5	4	3	2	1	0
Bit name	-	-	-	-	-	-	P2PU	P2M
Reset	0	0	0	0	0	0	1(0)	0(1)
Access	R	R	R	R	R	R	R/W	R/W

() are set when address/data multiplex mode.

Note that setting P2M to "0" in address/data multiplex mode is prohibited.

15.4.3 Pin Configurations

Table 15-4-1 shows the pin configurations for port 2.

Table 15-4-1 Port 2 Configuration

Port	Pin No.	P2n	P2M = "1"		P2M = "0"	
			P2nD = "1"	P2nD = "0"		
Port 2	87	P20	General-purpose output port	General-purpose input port	D8 *1	Data input/output
	84	P21	General-purpose output port	General-purpose input port	D9 *1	Data input/output
	83	P22	General-purpose output port	General-purpose input port	D10 *1	Data input/output
	82	P23	General-purpose output port	General-purpose input port	D11 *1	Data input/output
	81	P24	General-purpose output port	General-purpose input port	D12 *1	Data input/output
	80	P25	General-purpose output port	General-purpose input port	D13 *1	Data input/output
	78	P26	General-purpose output port	General-purpose input port	D14 *1	Data input/output
	77	P27	General-purpose output port	General-purpose input port	D15 *1	Data input/output

[Note 1]

▨ : When reset (in address/data separate mode)
"General-purpose input port" is selected in address/data multiplex mode.

*1 : In the event of a reset in address/data separate mode, the P2PU bit in the P2MD register is set to "1" and the data pins (the 8 bits D[15:8]) are pulled up.

[Note 2]

Setting P2M to "0" in address/data multiplex mode, is prohibited.

[Note 3]

When the bus authority is granted, D15 to D8 go to high impedance.

15.5 Port 3

15.5.1 Block Diagram

Fig. 15-5-1 shows a block diagram for port 3.

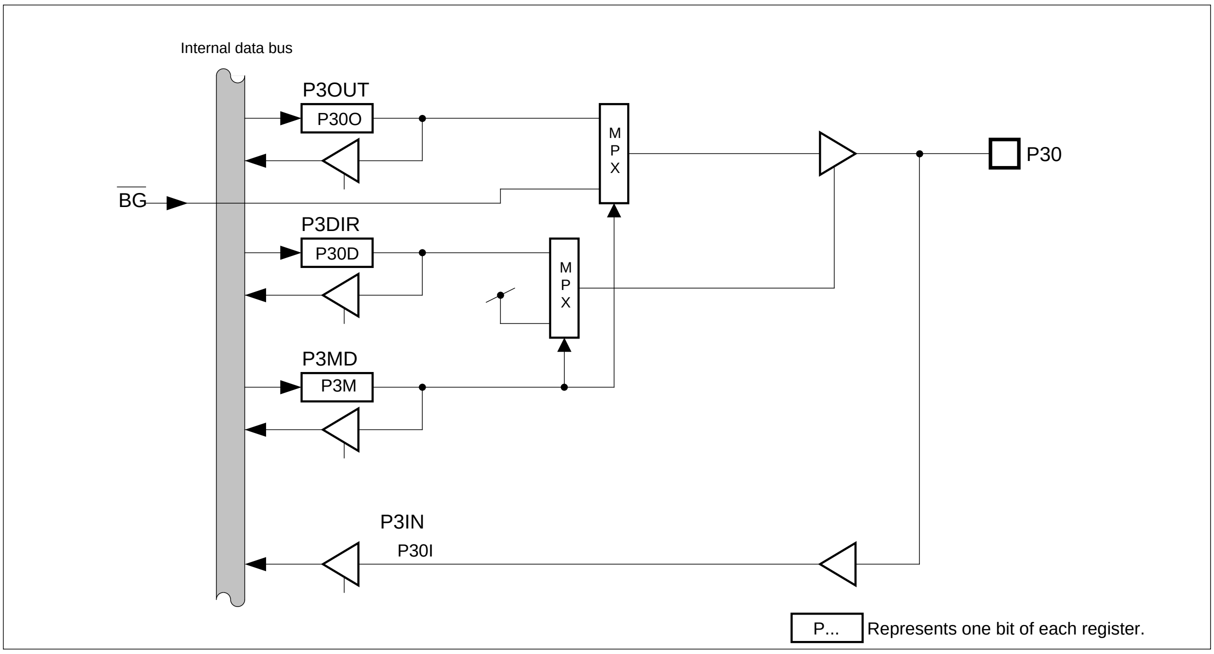


Fig. 15-5-1 Port 3 Block Diagram (P30)

15.5.2 Register Descriptions

Port 3 is a general-purpose input/output port that is also used for the bus grant signal $\overline{\text{BG}}$.

Each register for port 3 is described below.

Port 3 output register

Register symbol: P3OUT
Address: x'36008005
Purpose: This register sets the data to be output on port 3.

Bit No.	7	6	5	4	3	2	1	0
Bit name	-	-	-	-	-	-	-	P30O
Reset	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R/W

Port 3 pin register

Register symbol: P3IN
Address: x'36008085
Purpose: This register reads the value of the port 3 pin.

Bit No.	7	6	5	4	3	2	1	0
Bit name	-	-	-	-	-	-	-	P30I
Reset	0	0	0	0	0	0	0	X
Access	R	R	R	R	R	R	R	R

Port 3 input/output control register

Register symbol: P3DIR
Address: x'36008065
Purpose: This register sets the port 3 pin for input or output.
(0: input; 1: output)

Bit No.	7	6	5	4	3	2	1	0
Bit name	-	-	-	-	-	-	-	P30D
Reset	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R/W

Port 3 output mode register

Register symbol: P3MD
Address: x'36008025
Purpose: This register selects the content output on the port 3 pin.

Bit No.	7	6	5	4	3	2	1	0
Bit name	-	-	-	-	-	-	-	P3M
Reset	0	0	0	0	0	0	0	1
Access	R	R	R	R	R	R	R	R/W

P3M 0: Bus grant signal output ($\overline{\text{BG}}$)
1: General-purpose input/output port (P30)

Note: When $\overline{\text{BG}}$ is selected in the P3MD register, this control signal is output regardless of the value in the P3DIR register.
The input/output settings for this general-purpose port are made through the P3DIR register.

15.5.3 Pin Configurations

Table 15-5-1 shows the pin configurations for port 3.

Table 15-5-1 Port 3 Configuration

Port	Pin No.	P3n	P3M = "1"		P3M = "0"	
			P30D = "1"	P30D = "0"		
Port 3	76	P30	General-purpose output port	General-purpose input port	$\overline{\text{BG}}$	Bus grant signal output

[Note]

▨ : When reset (in address/data separate mode or address/data multiplex mode)

15.6 Port 4

15.6.1 Block Diagram

Figs. 15-6-1 to 15-6-4 show block diagrams for port 4.

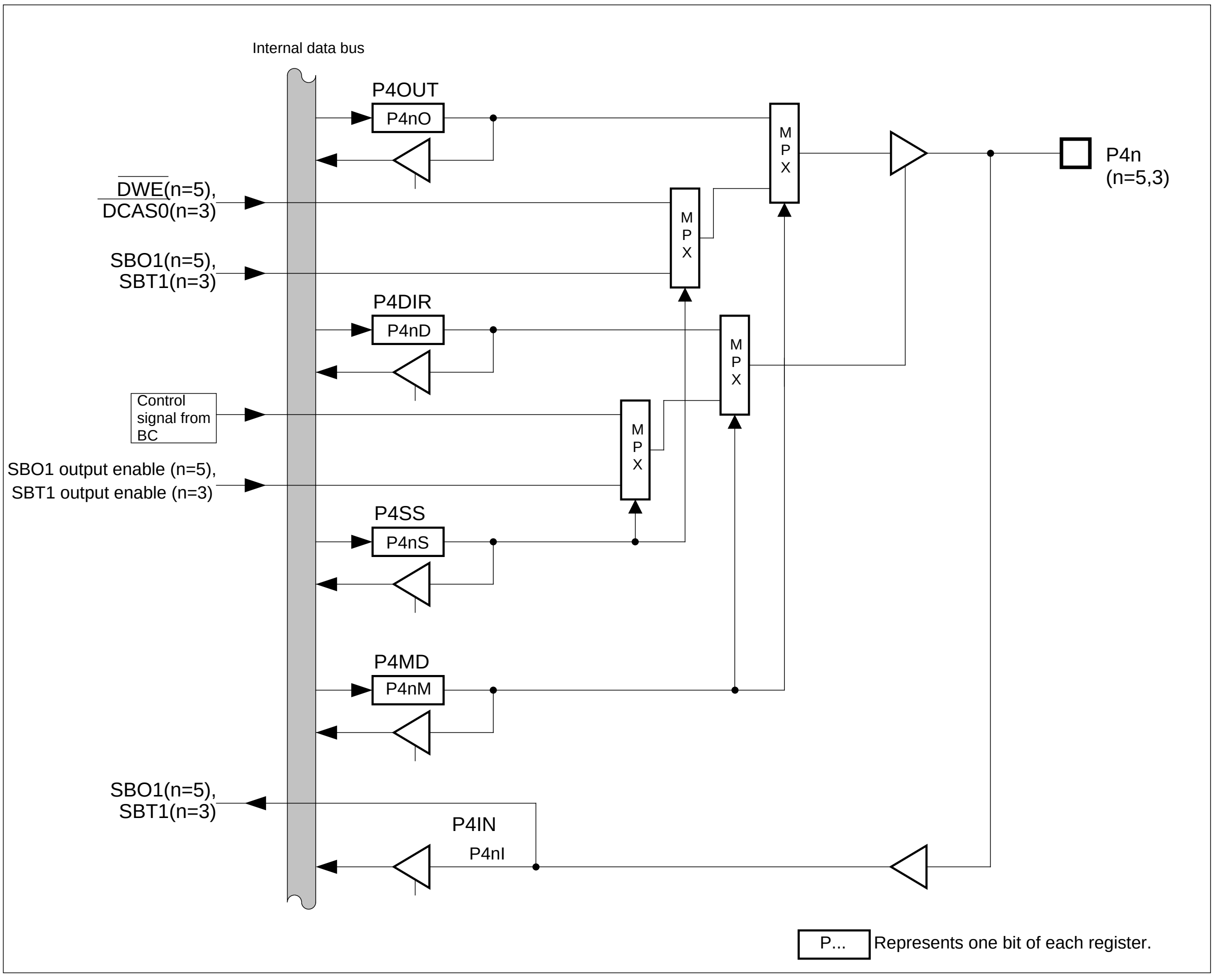


Fig. 15-6-1 Port 4 Block Diagram (P45 and P43)

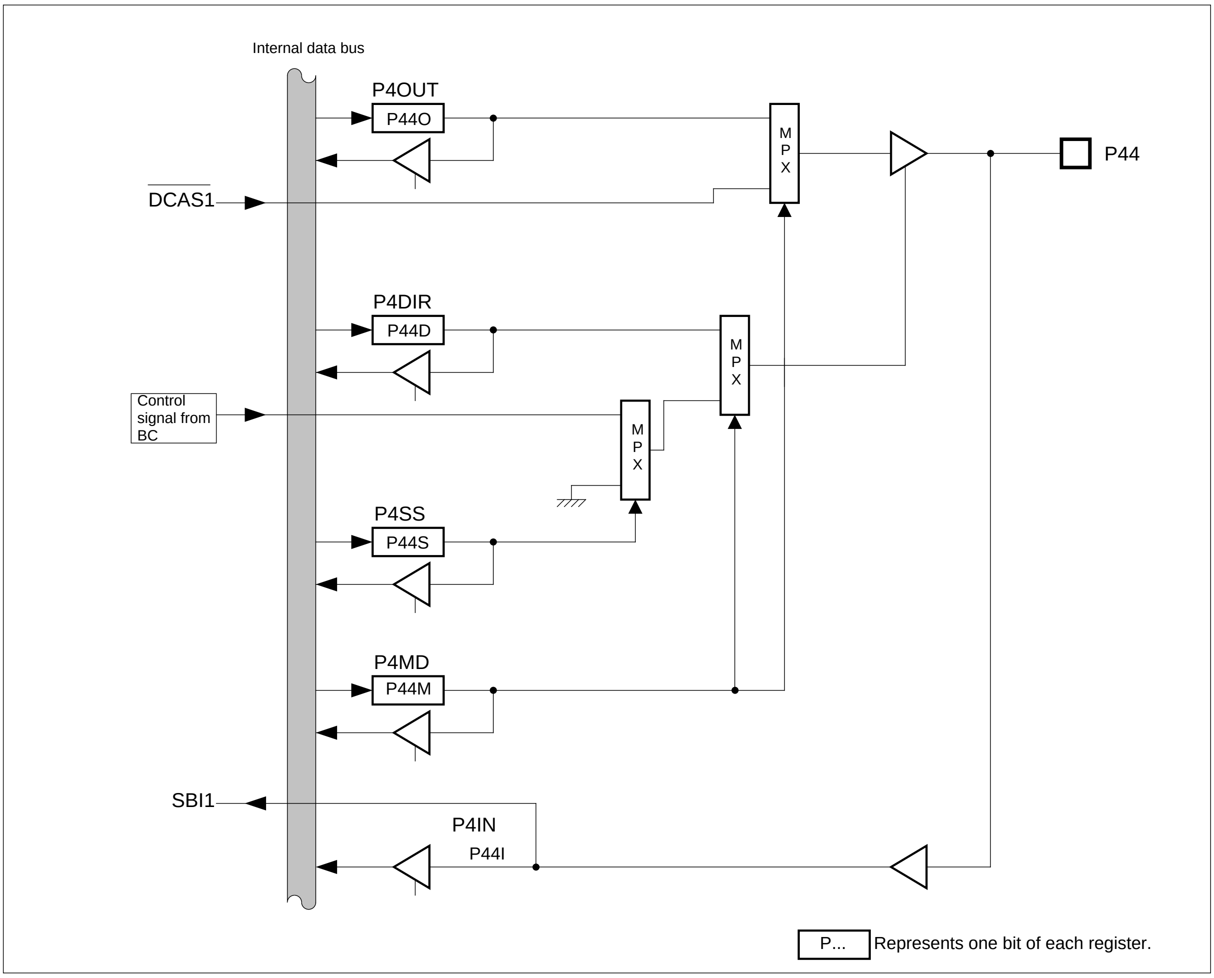


Fig. 15-6-2 Port 4 Block Diagram (P44)

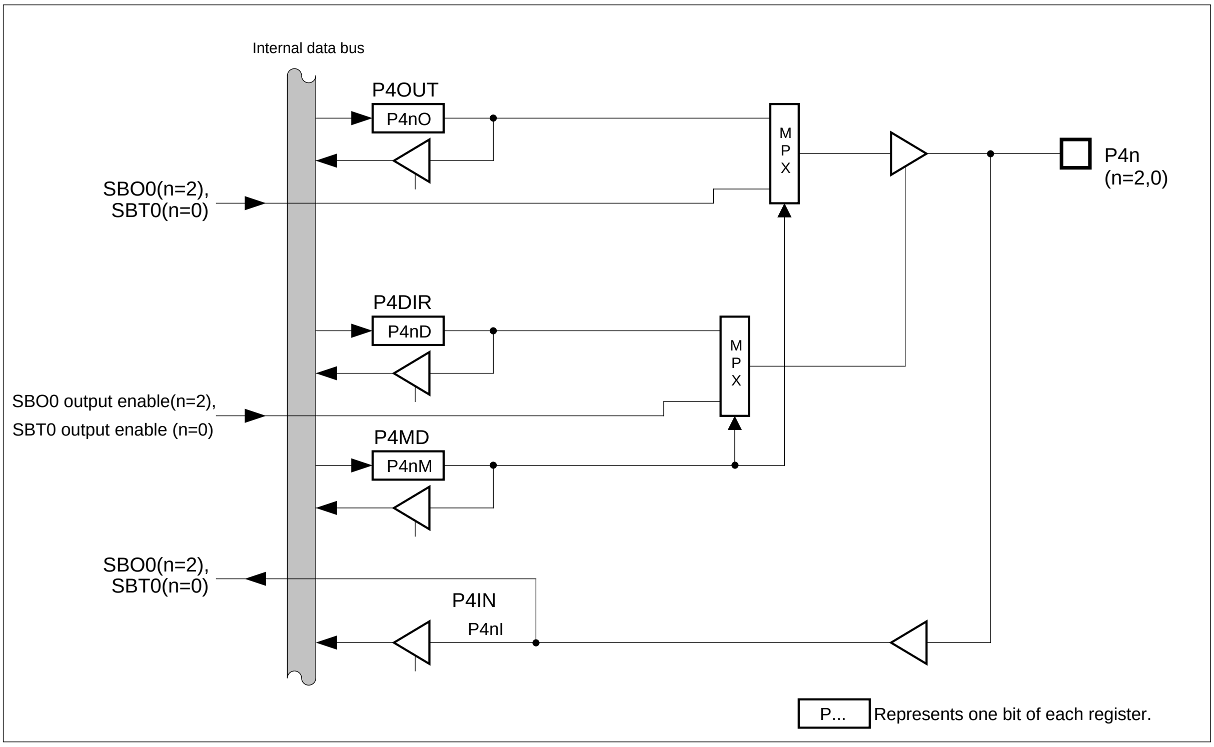


Fig. 15-6-3 Port 4 Block Diagram (P42, P40)

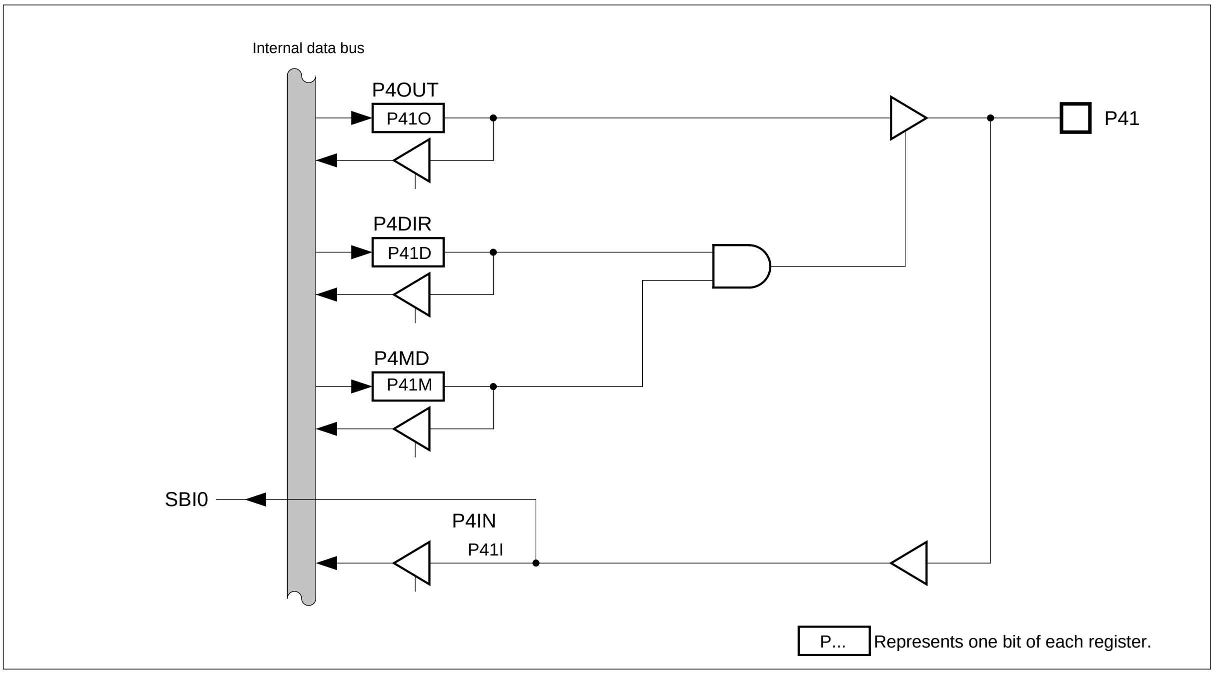


Fig. 15-6-4 Port 4 Block Diagram (P41)

15.6.2 Register Descriptions

Port 4 is a general-purpose input/output port that is also used for serial interface input/output signals SBI1, SBO1, SBT1, SBI0, SBO0, and SBT0; the DRAM CAS signals (for 2$\overline{\text{CAS}}$) $\overline{\text{DCAS1}}$ and $\overline{\text{DCAS0}}$; and the DRAM write signal (for 2$\overline{\text{CAS}}$) $\overline{\text{DWE}}$.

Each register for port 4 is described below.

Port 4 output register

Register symbol: P4OUT
Address: x'36008008
Purpose: This register sets the data to be output on port 4.

Bit No.	7	6	5	4	3	2	1	0
Bit name	-	-	P45O	P44O	P43O	P42O	P41O	P40O
Reset	0	0	0	0	0	0	0	0
Access	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Port 4 pin register

Register symbol: P4IN
Address: x'36008088
Purpose: This register reads the value of the port 4 pins.

Bit No.	7	6	5	4	3	2	1	0
Bit name	-	-	P45I	P44I	P43I	P42I	P41I	P40I
Reset	0	0	X	X	X	X	X	X
Access	R	R	R	R	R	R	R	R

Port 4 input/output control register

Register symbol: P4DIR
Address: x'36008068
Purpose: This register sets the port 4 pins for input or output.
(0: input; 1: output)

Bit No.	7	6	5	4	3	2	1	0
Bit name	-	-	P45D	P44D	P43D	P42D	P41D	P40D
Reset	0	0	0	0	0	0	0	0
Access	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Port 4 output mode register

Register symbol: P4MD
Address: x'36008028
Purpose: Along with P4SS, this register selects the content output on the port 4 pins.

Bit No.	7	6	5	4	3	2	1	0
Bit name	-	-	P45M	P44M	P43M	P42M	P41M	P40M
Reset	0	0	1	1	1	1	1	1
Access	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Port 4 dedicated output control register

Register symbol: P4SS
Address: x'36008048
Purpose: Along with P4MD, this register selects the content output on the port 4 pins.

Bit No.	7	6	5	4	3	2	1	0
Bit name	-	-	P45S	P44S	P43S	-	-	-
Reset	0	0	0	0	0	0	0	0
Access	R	R	R/W	R/W	R/W	R	R	R

P45M; P45S 00: Serial 1 data input/output (SBO1)
* The input/output setting is made through the serial interface 1 settings.
01: DRAM write signal output [for 2$\overline{\text{CAS}}$] ($\overline{\text{DWE}}$)
1x: General-purpose input/output port (P45)

P44M; P44S 00: Serial 1 data input (SBI1)
01: DRAM CAS signal 1 output [for 2$\overline{\text{CAS}}$] ($\overline{\text{DCAS1}}$)
1x: General-purpose input/output port (P44)

P43M; P43S 00: Serial 1 transfer clock input/output (SBT1)
* The input/output setting is made through the serial interface 1 settings.
01: DRAM CAS signal 0 output [for 2$\overline{\text{CAS}}$] ($\overline{\text{DCAS0}}$)
1x: General-purpose input/output port (P43)

P42M 0: Serial 0 data input/output (SBO0)
* The input/output setting is made through the serial interface 0 settings.
1: General-purpose input/output port (P42)

P41M 0: Serial 0 data input (SBI0)
1: General-purpose input/output port (P41)

P40M 0: Serial 1 transfer clock input/output (SBT0)
* The input/output setting is made through the serial interface 0 settings.
1: General-purpose input/output port (P40)

Note: If $\overline{\text{DWE}}$, $\overline{\text{DCAS1}}$, and $\overline{\text{DCAS0}}$ are selected in the P4MD and P4SS registers, each of these control signals is output, regardless of the value of P4DIR. The input/output settings for this general-purpose port are made through the P4DIR register.

15.6.3 Pin Configurations

Table 15-6-1 shows the pin configurations for port 4.

Table 15-6-1 Port 4 Configuration

Port	Pin	P4n	P4nM = "1"		P4nM = "0"			
	No.		P4nD = "1"	P4nD = "0"	P4nS = "1"		P4nS = "0"	
Port 4	75	P40	General-purpose output port	General-purpose input port	SBT0 [*1]	Serial 0 transfer clock input/output		
	74	P41	General-purpose output port	General-purpose input port	SBI0	Serial 0 data input		
	73	P42	General-purpose output port	General-purpose input port	SBO0 [*1]	Serial 0 data input/output		
	71	P43	General-purpose output port	General-purpose input port	$\overline{\text{DCAS0}}$	DRAM CAS signal 0 output (for 2$\overline{\text{CAS}}$)	SBT1 [*2]	Serial 1 transfer clock input/output
	70	P44	General-purpose output port	General-purpose input port	$\overline{\text{DCAS1}}$	DRAM CAS signal 1 output (for 2$\overline{\text{CAS}}$)	SBI1	Serial 1 data input
	68	P45	General-purpose output port	General-purpose input port	$\overline{\text{DWE}}$	DRAM write signal output (for 2$\overline{\text{CAS}}$)	SBO1 [*2]	Serial 1 data input/output

[Note 1]

▨ : When reset (whether in address/data separate mode or address/data multiplex mode)

*1 : The input/output setting is made through the serial interface 0 settings.

*2 : The input/output setting is made through the serial interface 1 settings.

Note: For details on the serial interface settings, refer to chapter 13, "Serial Interface."

[Note 2]

When the bus authority is granted, $\overline{\text{DWE}}$, $\overline{\text{DCAS1}}$, and $\overline{\text{DCAS0}}$ go to high impedance.

15.7 Port 5

15.7.1 Block Diagram

Figs. 15-7-1 to 15-7-5 show block diagrams for port 5.

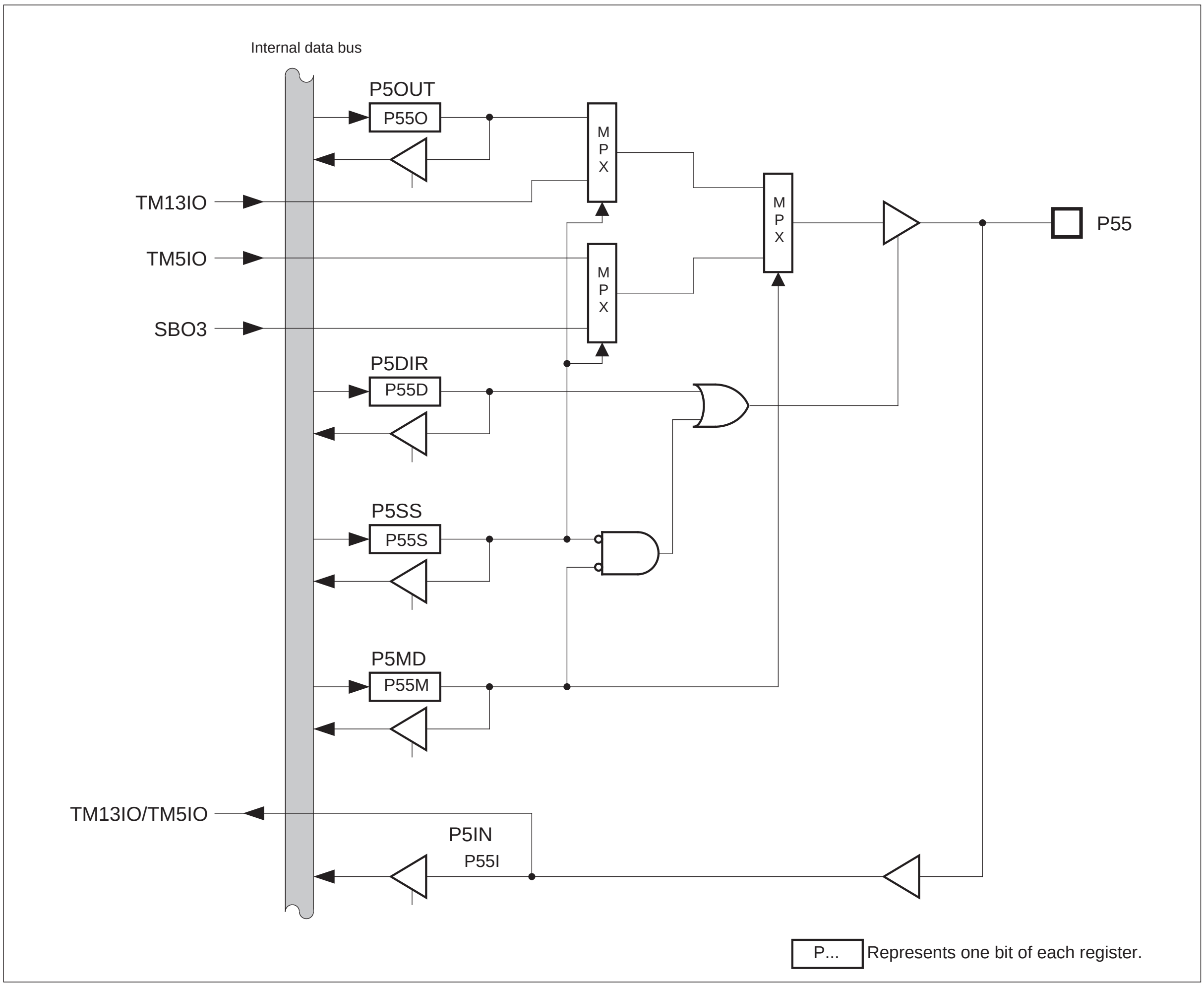


Fig. 15-7-1 Port 5 Block Diagram (P55)

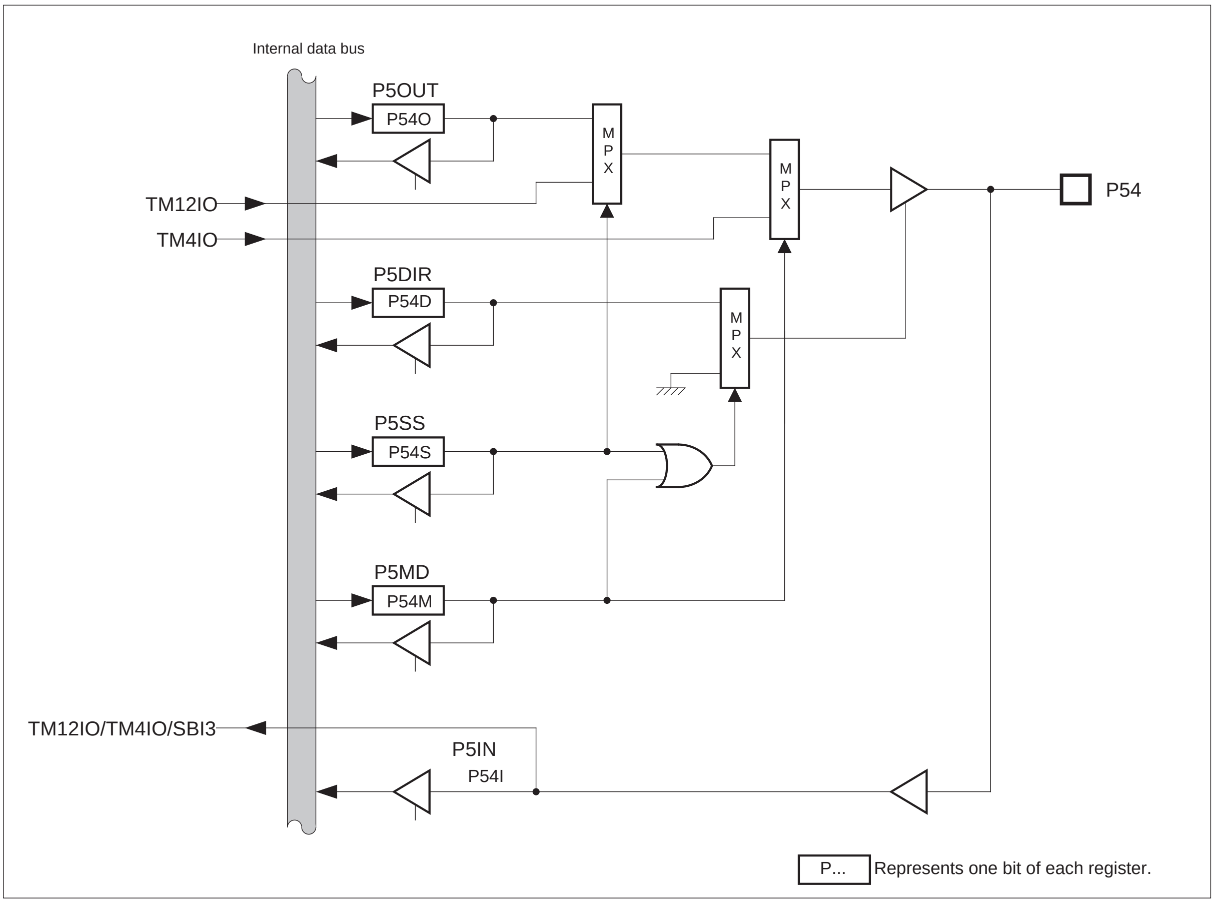


Fig. 15-7-2 Port 5 Block Diagram (P54)

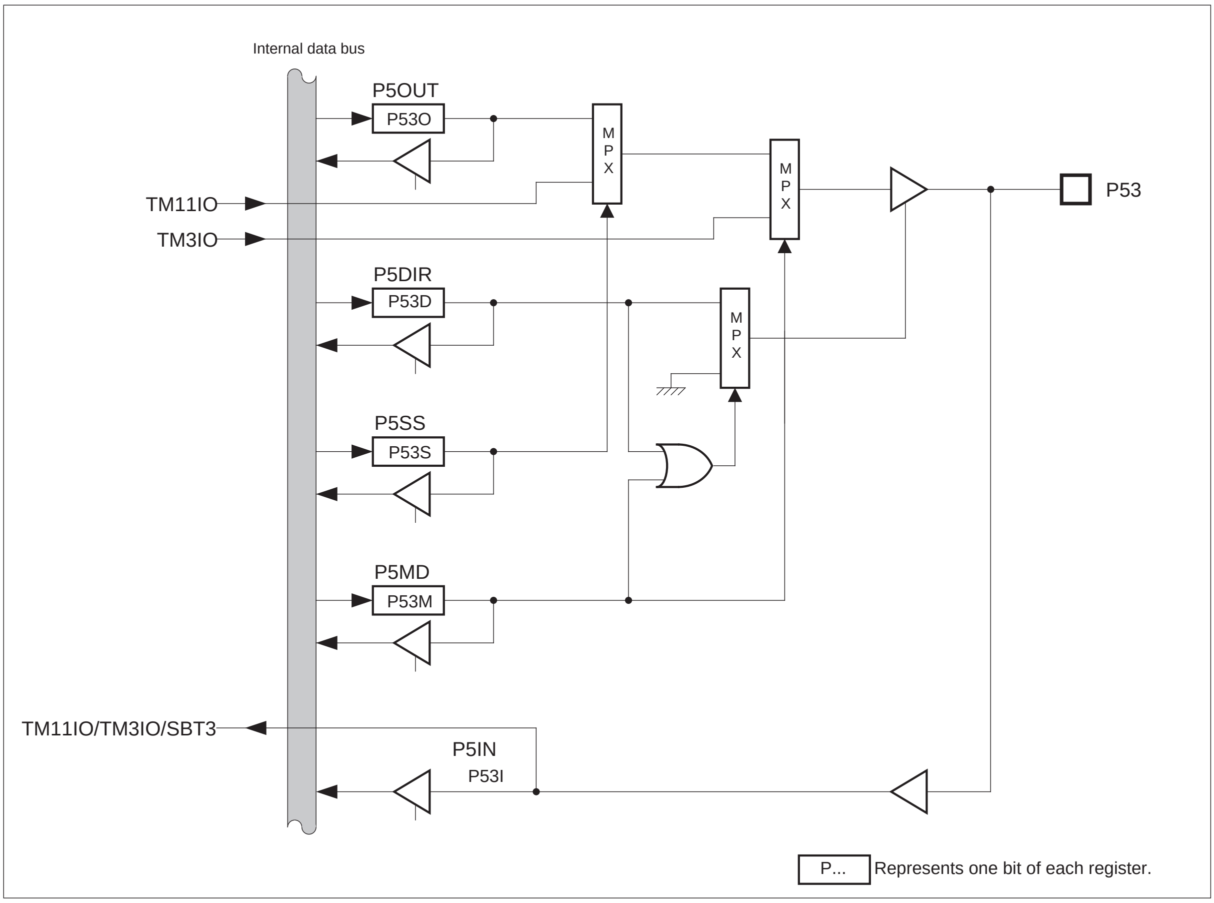


Fig. 15-7-3 Port 5 Block Diagram (P53)

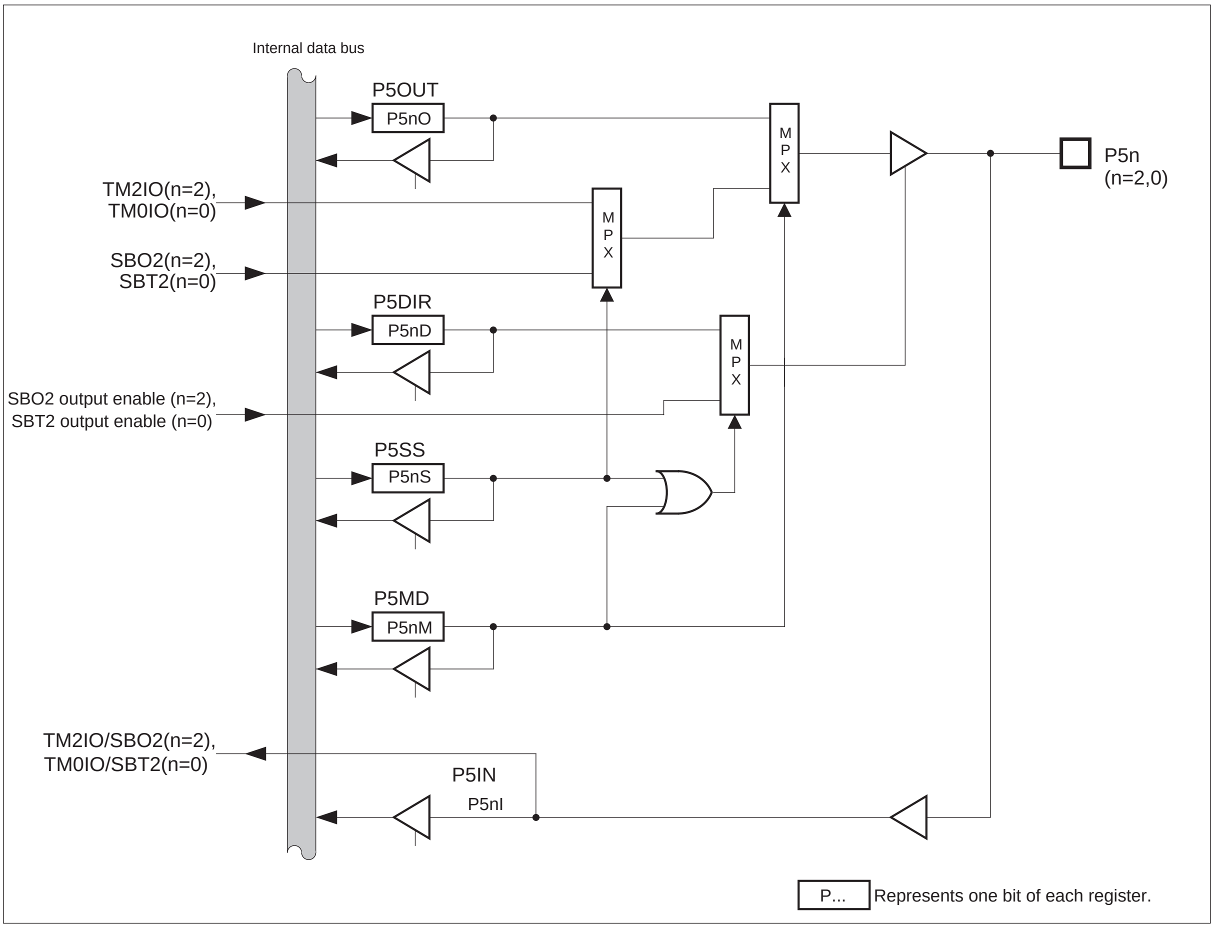


Fig. 15-7-4 Port 5 Block Diagram (P52, P50)

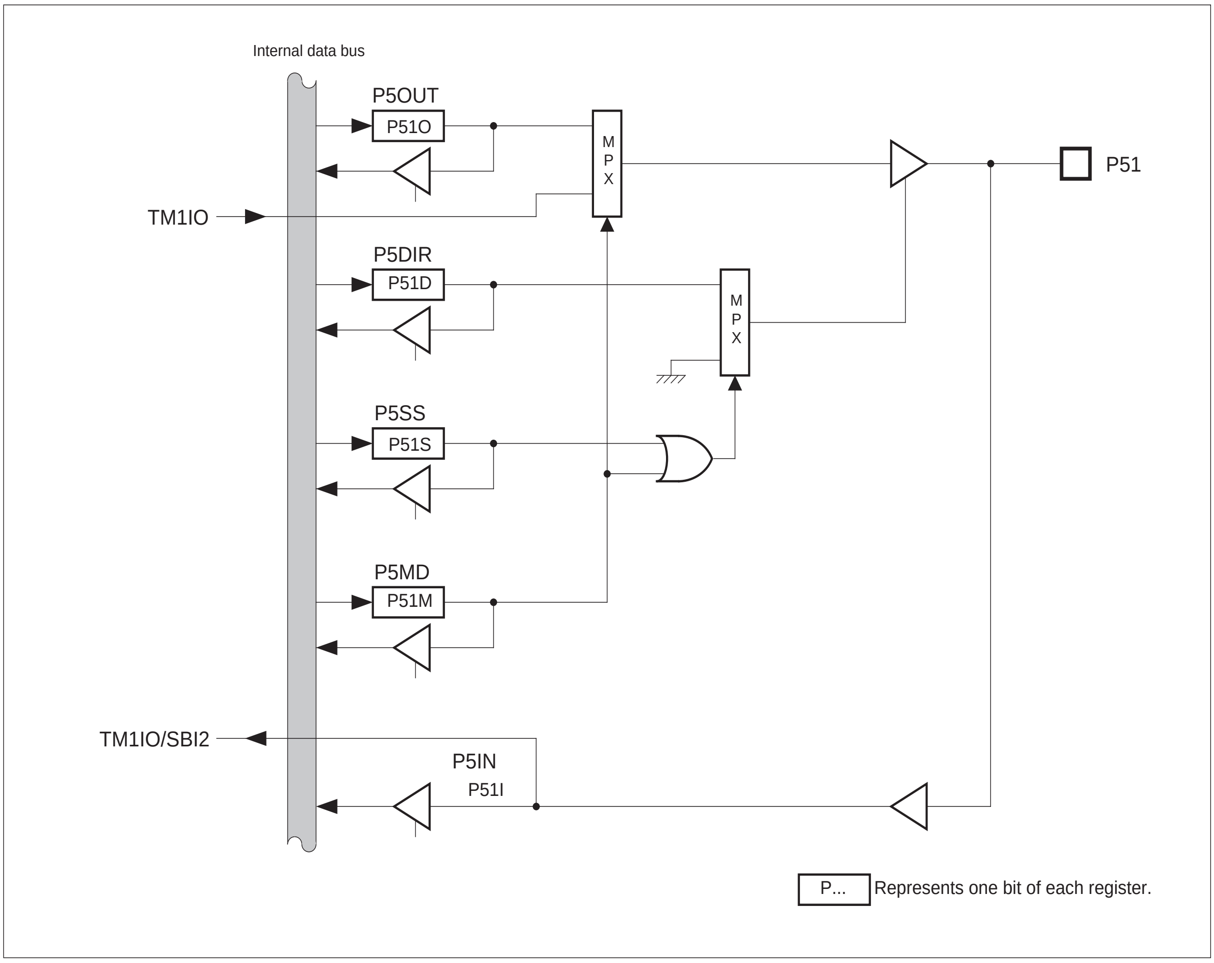


Fig. 15-7-5 Port 5 Block Diagram (P51)

15.7.2 Register Descriptions

Port 5 is a general-purpose input/output port that is also used for the serial interface input/output signals SBI3, SBO3, SBT3, SBI2, SBO2, SBT2; and the timer input/output signals TM13IO, TM12IO, TM11IO, TM5IO, TM4IO, TM3IO, TM2IO, TM1IO, and TM0IO.

Each register for port 5 is described below.

Port 5 output register

Register symbol: P5OUT
Address: x'36008009
Purpose: This register sets the data to be output on port 5.

Bit No.	7	6	5	4	3	2	1	0
Bit name	-	-	P55O	P54O	P53O	P52O	P51O	P50O
Reset	0	0	0	0	0	0	0	0
Access	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Port 5 pin register

Register symbol: P5IN
Address: x'36008089
Purpose: This register reads the value of the port 5 pins.

Bit No.	7	6	5	4	3	2	1	0
Bit name	-	-	P55I	P54I	P53I	P52I	P51I	P50I
Reset	0	0	X	X	X	X	X	X
Access	R	R	R	R	R	R	R	R

Port 5 input/output control register

Register symbol: P5DIR
Address: x'36008069
Purpose: This register sets the port 5 pins for input or output.
(0: input; 1: output)

Bit No.	7	6	5	4	3	2	1	0
Bit name	-	-	P55D	P54D	P53D	P52D	P51D	P50D
Reset	0	0	0	0	0	0	0	0
Access	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Port 5 output mode register

Register symbol: P5MD
Address: x'36008029
Purpose: Along with P5SS, this register selects the content output on the port 5 pins.

Bit No.	7	6	5	4	3	2	1	0
Bit name	-	-	P55M	P54M	P53M	P52M	P51M	P50M
Reset	0	0	1	1	1	1	1	1
Access	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Port 5 dedicated output control register

Register symbol: P5SS
Address: x'36008049
Purpose: Along with P5MD, this register selects the content output on the port 5 pins.

Bit No.	7	6	5	4	3	2	1	0
Bit name	-	-	P55S	P54S	P53S	P52S	P51S	P50S
Reset	0	0	1	1	1	1	1	1
Access	R	R	R/W	R/W	R/W	R/W	R/W	R/W

P55M; P55S 00: Serial 3 data output (SBO3)
01: Timer input/output (TM5IO) * 8-bit timer
10: Timer input/output (TM13IO) * 16-bit timer
11: General-purpose input/output port (P55)

P54M; P54S 00: Serial 3 data input (SBI3)
01: Timer input/output (TM4IO) * 8-bit timer
10: Timer input/output (TM12IO) * 16-bit timer
11: General-purpose input/output port (P54)

P53M; P53S 00: Serial 3 transfer clock input (SBT3)
01: Timer input/output (TM3IO) * 8-bit timer
10: Timer input/output (TM11IO) * 16-bit timer
11: General-purpose input/output port (P53)

P52M; P52S 00: Serial 2 data input/output (SBO2)
* The input/output settings depend on the serial interface 2 settings and the timing.
01: Timer input/output (TM2IO) * 8-bit timer
1x: General-purpose input/output port (P52)

P51M; P51S 00: Serial 2 data input (SBI2)
01: Timer input/output (TM1IO) * 8-bit timer
1x: General-purpose input/output port (P51)

P50M; P50S 00: Serial 2 transfer clock input/output (SBT2)
* The input/output settings depend on the serial interface 2 settings and the timing.
01: Timer input/output (TM0IO) * 8-bit timer
1x: General-purpose input/output port (P50)

Note: The input/output settings for this general-purpose port and the timer are made through the P5DIR register.

15.7.3 Pin Configurations

Table 15-7-1 shows the pin configurations for port 5.

Table 15-7-1 Port 5 Configuration

Port	Pin No.	P5n	P5nM = "1"					P5nM = "0"				
			P5nS = "1"		P5nS = "0"			P5nS = "1"			P5nS = "0"	
			P5nD = "1"	P5nD = "0"	P5nD ="1"		P5nD ="0"	P5nD = "1"		P5nD = "0"		
Port 5	67	P50	General-purpose output port	General-purpose input port	P50	General-purpose output port	General-purpose input port	TM0IO	Timer 0 or timer 8 output *1	Timer 0 or timer 8 input	SBT2 *5	Serial 2 transfer clock input/ output
	66	P51	General-purpose output port	General-purpose input port	P51	General-purpose output port	General-purpose input port	TM1IO	Timer 1 or timer 9 output *2	Timer 1 or timer 9 input	SBI2	Serial 2 data input
	65	P52	General-purpose output port	General-purpose input port	P52	General-purpose output port	General-purpose input port	TM2IO	Timer 2 or timer A output *3	Timer 2 or timer A input	SBO2 *5	Serial 2 data input/output
	64	P53	General-purpose output port	General-purpose input port	TM11IO	Timer 11 output	Timer 11 input	TM3IO	Timer 3 or timer B output *4	Timer 3 or timer B input	SBT3 *6	Serial 3 transfer clock input
	63	P54	General-purpose output port	General-purpose input port	TM12IO	Timer 12 output	Timer 12 input	TM4IO	Timer 4 output	Timer 4 input	SBI3	Serial 3 data input
	62	P55	General-purpose output port	General-purpose input port	TM13IO	Timer 13 output	Timer 13 input	TM5IO	Timer 5 output	Timer 5 input	SBO3	Serial 3 data output

[Note]

▨	: When reset (whether in address/data separate mode or address/data multiplex mode)
*1 to *4	: Set the respective output selections for timer 3/timer B output, timer 2/timer A output, timer 1/timer 9 output, and timer 0/timer 8 output in the 8-bit timer TMOSL register.
*5	: The input/output settings depend on the serial interface 2 settings and the timing.
*6	: When serial 3 transfer clock input is selected, the P53D bit in the P5DIR register must be set to "0".

Note: For details on the TMOSL register, refer to section 10.5, "Description of Registers."
For details on the input/output settings for the serial interface pins, refer to "Description of Registers" in chapter 13.

15.8 Port 6

15.8.1 Block Diagram

Figs. 15-8-1 shows the block diagrams for port 6.

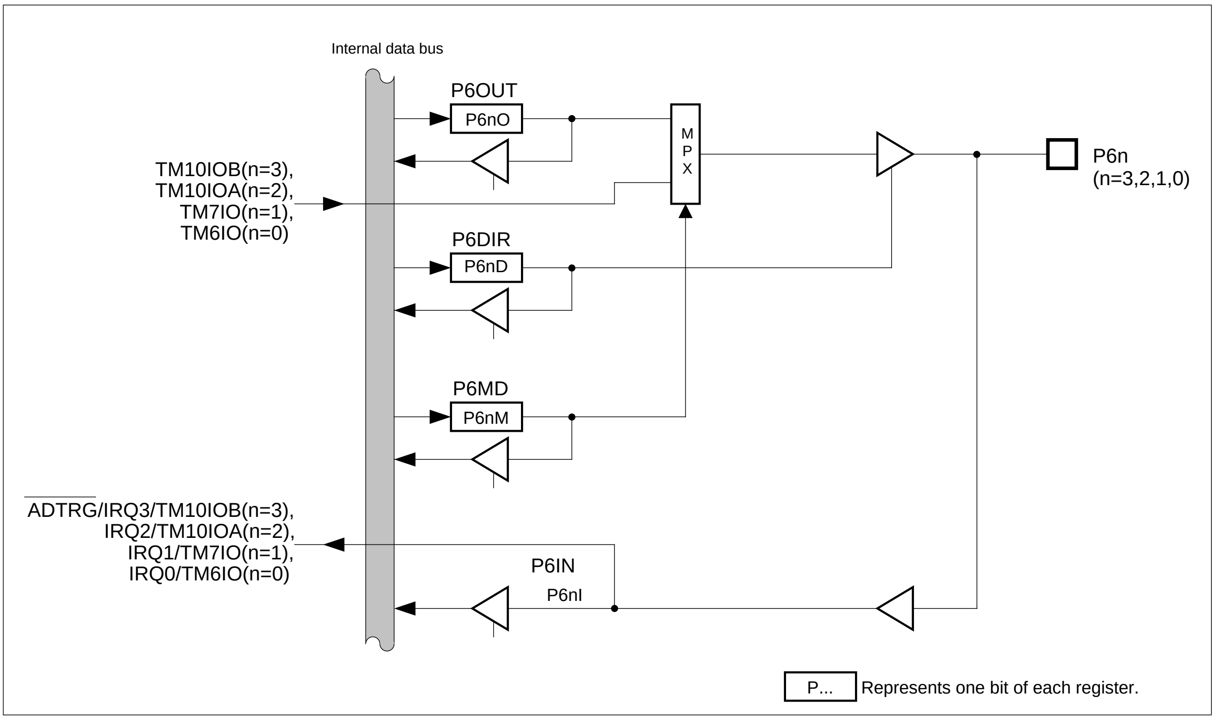


Fig. 15-8-1 Port 6 Block Diagram (P63 to P60)

15.8.2 Register Descriptions

Port 6 is a general-purpose input/output port that is also used for external interrupt inputs IRQ3 to $\overline{\text{IRQ0}}$; the timer input/output signals TM6IO, TM7IO, TM10IOA, TM10IOB; and the A/D conversion trigger input $\overline{\text{ADTRG}}$.

Each register for port 6 is described below.

Port 6 output register

Register symbol: P6OUT
Address: x'3600800C
Purpose: This register sets the data to be output on port 6.

Bit No.	7	6	5	4	3	2	1	0
Bit name	-	-	-	-	P63O	P62O	P61O	P60O
Reset	0	0	0	0	0	0	0	0
Access	R	R	R	R	R/W	R/W	R/W	R/W

Port 6 pin register

Register symbol: P6IN
Address: x'3600808C
Purpose: This register reads the value of the port 6 pins.

Bit No.	7	6	5	4	3	2	1	0
Bit name	-	-	-	-	P63I	P62I	P61I	P60I
Reset	0	0	0	0	X	X	X	X
Access	R	R	R	R	R	R	R	R

Port 6 input/output control register

Register symbol: P6DIR
Address: x'3600806C
Purpose: This register sets the port 6 pins for input or output.
(0: input; 1: output)

Bit No.	7	6	5	4	3	2	1	0
Bit name	-	-	-	-	P63D	P62D	P61D	P60D
Reset	0	0	0	0	0	0	0	0
Access	R	R	R	R	R/W	R/W	R/W	R/W

Port 6 output mode register

Register symbol: P6MD
Address: x'3600802C
Purpose: This register selects the content output on the port 6 pins.

Bit No.	7	6	5	4	3	2	1	0
Bit name	-	-	-	-	P63M	P62M	P61M	P60M
Reset	0	0	0	0	1	1	1	1
Access	R	R	R	R	R/W	R/W	R/W	R/W

When P6nM is "0", the timer input/output signal is selected. The input/output setting for the timer input/output signal is also changed by P6nD.

15.8.3 Pin Configurations

Table 15-8-1 shows the pin configurations for port 6.

Table 15-8-1 Port 6 Configuration

Port	Pin	P6n	P6nM = "1"		P6nM = "0"		
			P6nD = "1"	P6nD = "0"	P6nD = "1"		P6nD = "0"
Port 6	59	P60	General-purpose output port	General-purpose input port	TM6IO	Timer 6 output	Timer 6 input
	58	P61	General-purpose output port	General-purpose input port	TM7IO	Timer 7 output	Timer 7 input
	57	P62	General-purpose output port	General-purpose input port	TM10IOA	Timer 10 output A	Timer 10 input A
	56	P63	General-purpose output port	General-purpose input port	TM10IOB	Timer 10 output B	Timer 10 input B

[Note 1]

▨ : When reset (whether in address/data separate mode or address/data multiplex mode)

[Note 2]
When Pin No. 56 to 59, respectively, are set as external interrupt input pins (IRQ3 to 0), or if pin No. 56 is set as the A/D conversion trigger input pin ($\overline{\text{ADTRG}}$), the pins must be set as a general-purpose input port as shown in the above table.

15.9 Port 7

15.9.1 Block Diagram

Fig. 15-9-1 and Fig. 15-9-2 show block diagrams for port 7.

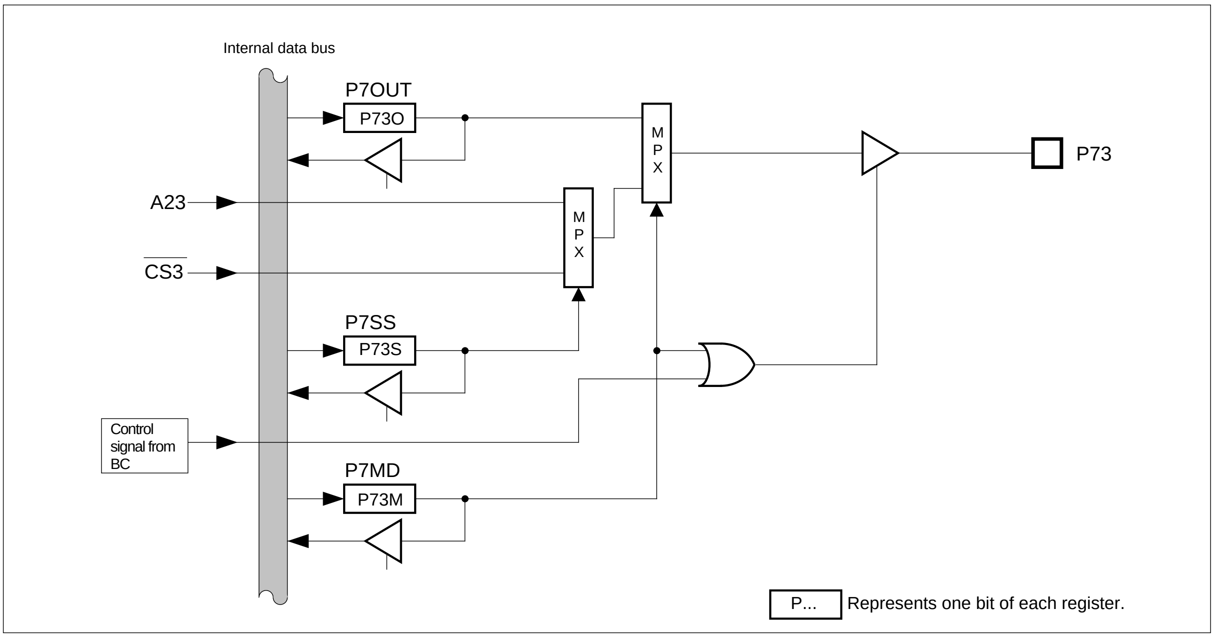


Fig. 15-9-1 Port 7 Block Diagram (P73)

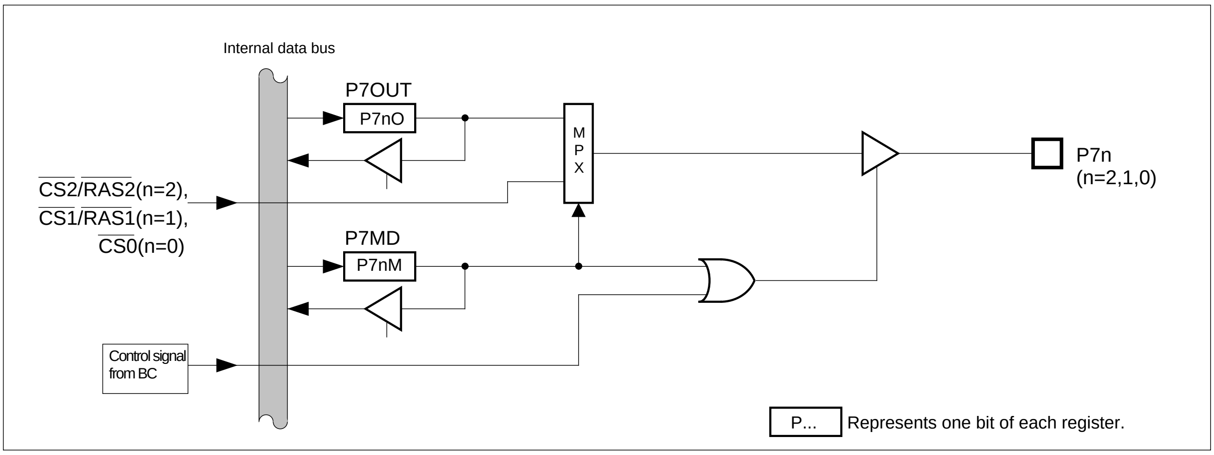


Fig. 15-9-2 Port 7 Block Diagram (P72 to P70)

15.9.2 Register Descriptions

Port 7 is a general-purpose output port that is also used for address bus signal A23, DRAM RAS signals $\overline{\text{RAS2}}$ and $\overline{\text{RAS1}}$, chip select signals $\overline{\text{CS3}}$ to $\overline{\text{CS0}}$.

Each register for port 7 is described below.

Port 7 output register

Register symbol: P7OUT
Address: x'3600800D
Purpose: This register sets the data to be output on port 7.

Bit No.	7	6	5	4	3	2	1	0
Bit name	-	-	-	-	P73O	P72O	P71O	P70O
Reset	0	0	0	0	0	0	0	0
Access	R	R	R	R	R/W	R/W	R/W	R/W

Port 7 output mode register

Register symbol: P7MD
Address: x'3600802D
Purpose: This register selects the content output on the port 7 pins.

Bit No.	7	6	5	4	3	2	1	0
Bit name	-	-	-	-	P73M	P72M	P71M	P70M
Reset	0	0	0	0	0	0	0	0
Access	R	R	R	R	R/W	R/W	R/W	R/W

Port 7 dedicated output control register

Register symbol: P7SS
Address: x'3600804D
Purpose: This register selects the content output on the port 7 pins.
Valid when the P7nM is “0”.

Bit No.	7	6	5	4	3	2	1	0
Bit name	-	-	-	-	P73S	-	-	-
Reset	0	0	0	0	0	0	0	0
Access	R	R	R	R	R/W	R	R	R

P73M; P73S 00: Chip select signal 3 output ($\overline{\text{CS3}}$)
01: Address output (A23)
1x: General-purpose output port (P73)

P72M 0: Chip select signal 2 output/DRAM RAS signal 2 output ($\overline{\text{CS2}}$/$\overline{\text{RAS2}}$)
* The $\overline{\text{CS2}}$/$\overline{\text{RAS2}}$ selection depends on the setting of the registers within the bus controller.
1: General-purpose output port (P72)

Note: For details on the bus controller register settings, refer to section 8.6, "Description of Registers."

P71M 0: Chip select signal 1 output/DRAM RAS signal 1 output ($\overline{\text{CS1}}$/$\overline{\text{RAS1}}$)
* The $\overline{\text{CS1}}$/$\overline{\text{RAS1}}$ selection depends on the setting of the registers within the bus controller.
1: General-purpose output port (P71)

P70M 0: Chip select signal 0 output ($\overline{\text{CS0}}$)
1: General-purpose output port (P70)

15.9.3 Pin Configurations

Table 15-9-1 shows the pin configurations for port 7.

Table 15-9-1 Port 7 Configuration

Port	Pin No.	P7n	P7nM = "1"	P7nM = "0"			
				P73S = "1"		P73S = "0"	
Port 7	55	P70	General-purpose output port	$\overline{\text{CS0}}$	Chip select signal 0 output		
	53	P71	General-purpose output port	$\overline{\text{CS1}}$ or $\overline{\text{RAS1}}$ *1	Chip select signal 1 output or DRAM RAS signal 1 output		
	52	P72	General-purpose output port	$\overline{\text{CS2}}$ or $\overline{\text{RAS2}}$ *2	Chip select signal 2 output or DRAM RAS signal 2 output		
	51	P73	General-purpose output port	A23	Address output	$\overline{\text{CS3}}$	Chip select signal 3 output

[Note 1]

▨ : When reset (whether in address/data separate mode or address/data multiplex mode)

*1 : If block 1 in the external memory space is not used as a DRAM space, $\overline{\text{CS1}}$ is selected; if block 1 is used as a DRAM space, $\overline{\text{RAS1}}$ is selected.

*2 : If block 2 in the external memory space is not used as a DRAM space, $\overline{\text{CS2}}$ is selected; if block 2 is used as a DRAM space, $\overline{\text{RAS2}}$ is selected.

Note: For details on the external memory space settings, refer to section 8.6, "Description of Registers."

[Note 2]

When the bus authority is granted, A23, $\overline{\text{CS3}}$, $\overline{\text{CS2}}$, $\overline{\text{RAS2}}$, $\overline{\text{CS1}}$, $\overline{\text{RAS1}}$, and $\overline{\text{CS0}}$ go to high impedance.

15.10 Port 8

15.10.1 Block Diagram

Figs. 15-10-1 shows the block diagrams for port 8.

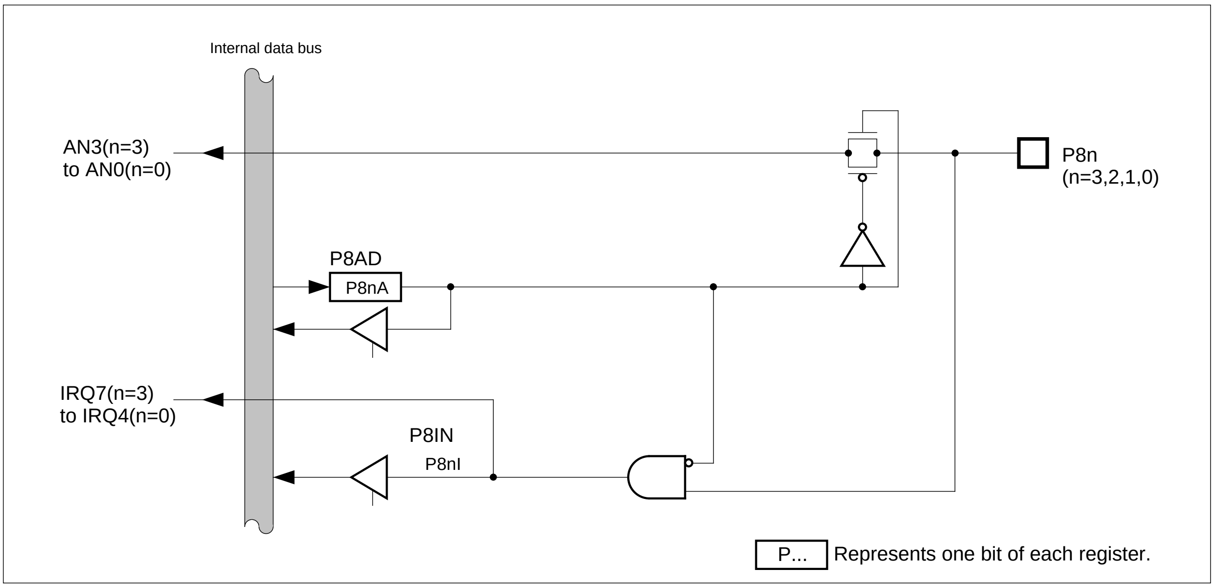


Fig. 15-10-1 Port 8 Block Diagram (P83 to P80)

15.10.2 Register Descriptions

Port 8 is a general-purpose input port that is also used for analog signal inputs AN3 to AN0 and external interrupt inputs IRQ7 to IRQ4.

Each register for port 8 is described below.

Port 8 analog/digital input control register

Register symbol: P8AD
Address: x'36008030
Purpose: This register selects the analog/digital input on the port 8 pins.
(0: digital; 1: analog)

Bit No.	7	6	5	4	3	2	1	0
Bit name	-	-	-	-	P83A	P82A	P81A	P80A
Reset	0	0	0	0	1	1	1	1
Access	R	R	R	R	R/W	R/W	R/W	R/W

Port 8 pin register

Register symbol: P8IN
Address: x'36008090
Purpose: This register reads the value of the port 8 pins.

Bit No.	7	6	5	4	3	2	1	0
Bit name	-	-	-	-	P83I	P82I	P81I	P80I
Reset	0	0	0	0	X	X	X	X
Access	R	R	R	R	R	R	R	R

When P8nA is "1", reading this register returns a value of "0", regardless of the actual values of the port pins. When P8nA is "1", regardless of the actual values of the port pins, IRQ7 to IRQ4 are treated as if they were "L" by the microcontroller internally.

15.10.3 Pin Configurations

Table 15-10-1 shows the pin configurations for port 8.

Table 15-10-1 Port 8 Configuration

Port	Pin No.	P8n	P8nA = "0"	P8nA = "1"	
Port 8	48	P80	General-purpose input port	AN0	Analog signal input
	47	P81	General-purpose input port	AN1	Analog signal input
	46	P82	General-purpose input port	AN2	Analog signal input
	45	P83	General-purpose input port	AN3	Analog signal input

[Note 1]

▨ : When reset (whether in address/data separate mode or address/data multiplex mode)

[Note 2]

When pin Nos. 45 to 48, respectively, are set as external interrupt input pins (IRQ7 to IRQ4), the pins must be set as a general-purpose input port as shown in the above table.

15.11 Port 9

15.11.1 Block Diagram

Fig. 15-11-1 to Fig. 15-11-4 show block diagrams for port 9.

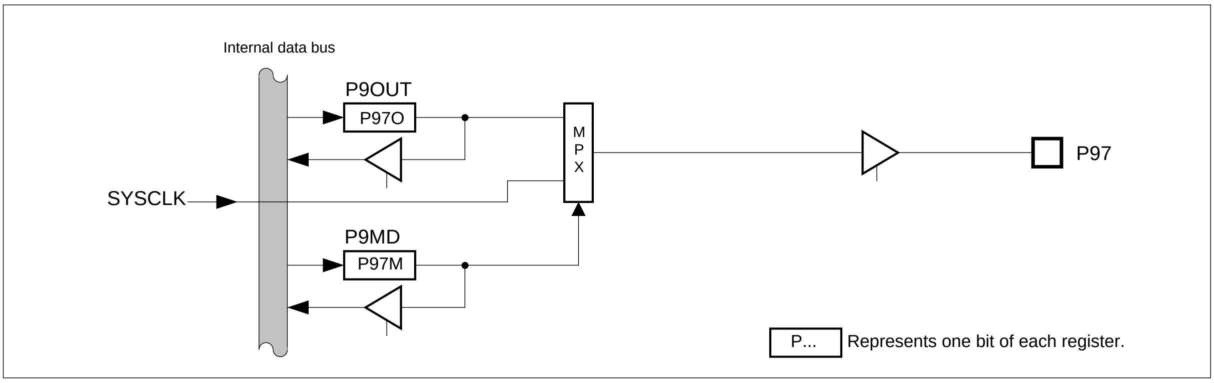


Fig. 15-11-1 Port 9 Block Diagram (P97)

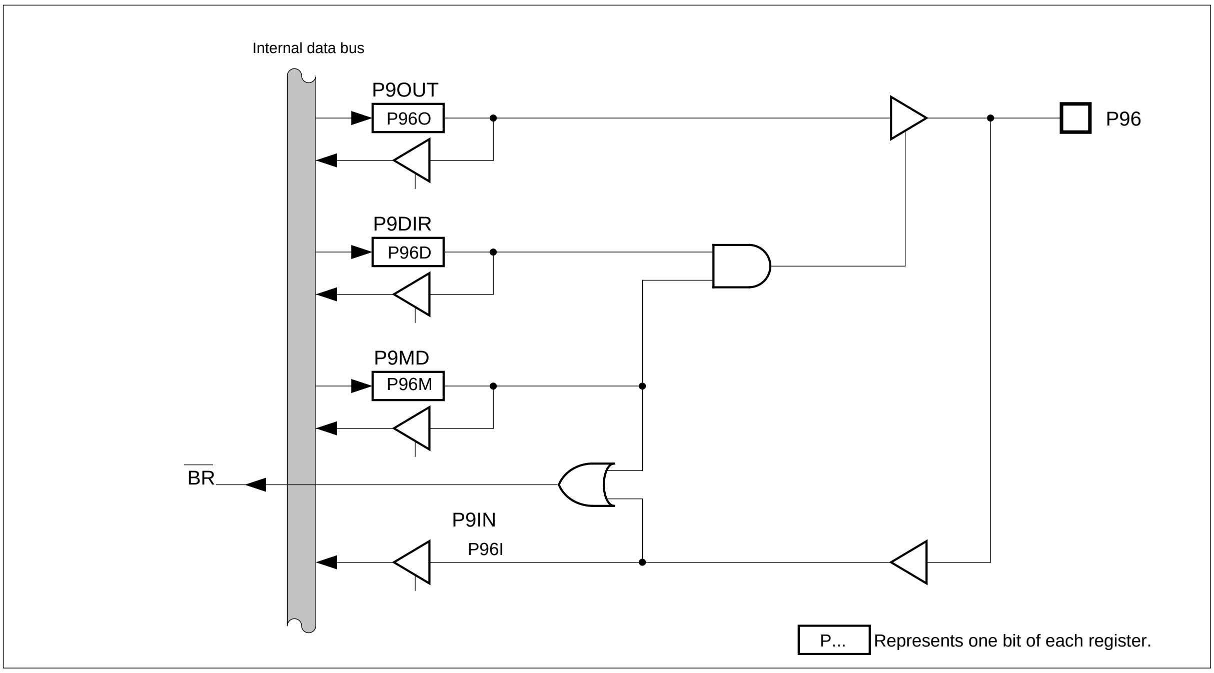


Fig. 15-11-2 Port 9 Block Diagram (P96)

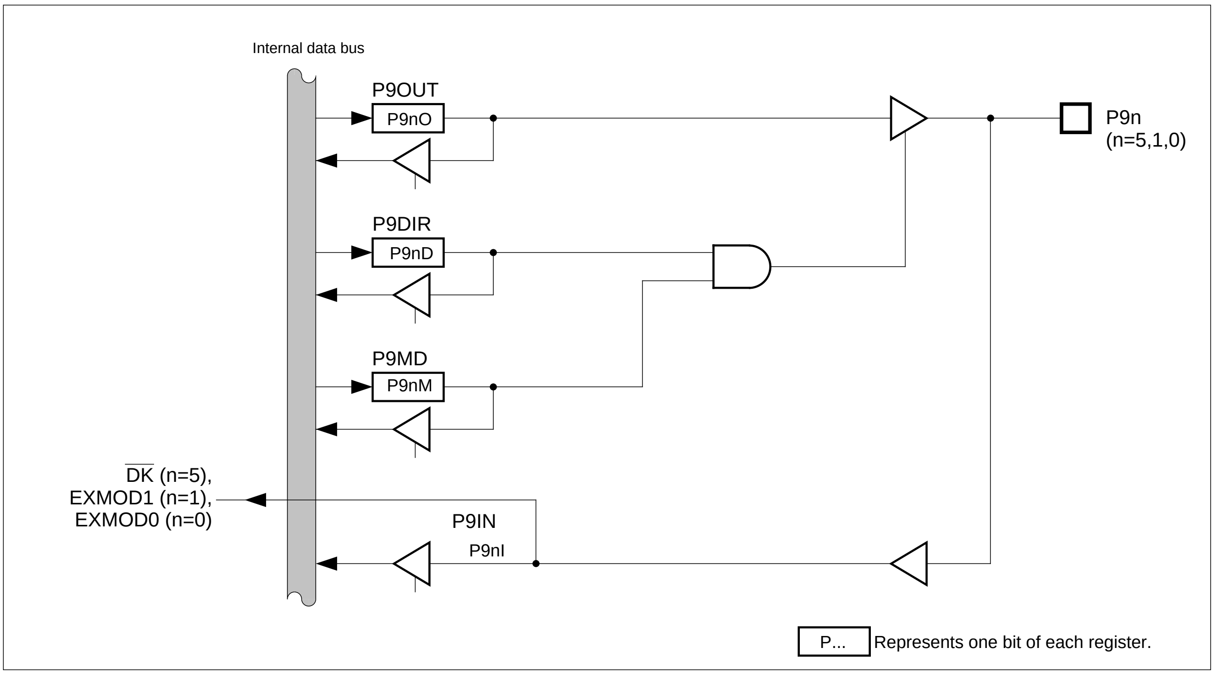


Fig. 15-11-3 Port 9 Block Diagram (P95, P91, P90)

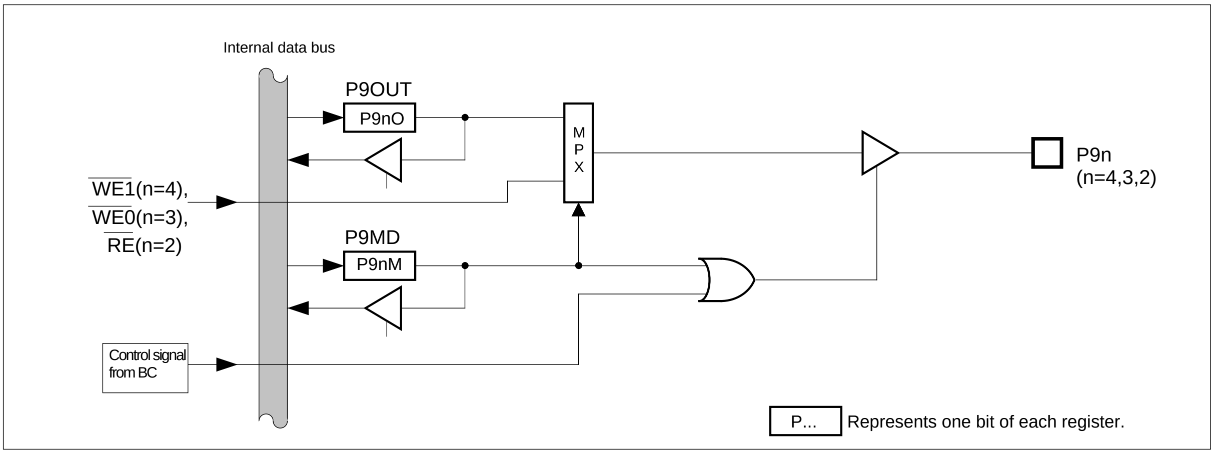


Fig. 15-11-4 Port 9 Block Diagram (P94, P93, P92)

15.11.2 Register Descriptions

Port 9 is also used for extension mode setting signals EXMOD1 and EXMOD0; memory write signals $\overline{\text{WE1}}$ and $\overline{\text{WE0}}$; memory read signal $\overline{\text{RE}}$; bus authority request signal $\overline{\text{BR}}$; data acknowledge signal $\overline{\text{DK}}$; and system clock SYSCLK. P96, P95, P91, and P90 are general-purpose input/output ports, and P97 and P94 to P92 are general-purpose output ports.

Each register for port 9 is described below.

Port 9 output register

Register symbol: P9OUT
Address: x'36008011
Purpose: This register sets the data to be output on port 9.

Bit No.	7	6	5	4	3	2	1	0
Bit name	P97O	P96O	P95O	P94O	P93O	P92O	P91O	P90O
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Port 9 pin register

Register symbol: P9IN
Address: x'36008091
Purpose: This register reads the value of the port 9 pins.

Bit No.	7	6	5	4	3	2	1	0
Bit name	-	P96I	P95I	-	-	-	P91I	P90I
Reset	0	X	X	0	0	0	X	X
Access	R	R	R	R	R	R	R	R

Port 9 input/output control register

Register symbol: P9DIR
Address: x'36008071
Purpose: This register sets the port 9 pins for input or output.
Valid when the P9nM is "1".
(0: input; 1: output)

Bit No.	7	6	5	4	3	2	1	0
Bit name	-	P96D	P95D	-	-	-	P91D	P90D
Reset	1	0	0	1	1	1	0	0
Access	R	R/W	R/W	R	R	R	R/W	R/W

Port 9 output mode register

Register symbol: P9MD
Address: x'36008031
Purpose: This register selects the content output on the port 9 pins.

Bit No.	7	6	5	4	3	2	1	0
Bit name	P97M	P96M	P95M	P94M	P93M	P92M	P91M	P90M
Reset	0	1	1	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P97M 0: System clock output (SYSCLK)
1: General-purpose output port (P97)

P96M 0: Bus authority request signal ($\overline{\text{BR}}$)
1: General-purpose input/output port (P96)

P95M 0: Data acknowledge signal input ($\overline{\text{DK}}$)
1: General-purpose input/output port (P95)

P94M 0: Memory write signal output ($\overline{\text{WE1}}$)
1: General-purpose output port (P94)

P93M 0: Memory write signal output ($\overline{\text{WE0}}$)
1: General-purpose output port (P93)

P92M 0: Memory read signal output ($\overline{\text{RE}}$)
1: General-purpose output port (P92)

P91M 1: General-purpose input/output port (P91)

P90M 1: General-purpose input/output port (P90)

Note: If $\overline{\text{BR}}$ and $\overline{\text{DK}}$ are selected in the P9MD register, these respective control signals are input regardless of the value of P9DIR.
The input/output settings for this general-purpose port is made through the P9DIR register.

15.11.3 Pin Configurations

Table 15-11-1 shows the pin configurations for port 9.

Table 15-11-1 Port 9 Configuration

Port	Pin No.	P9n	P9nM = "1"		P9nM = "0"	
			P9nD = "1"	P9nD = "0"		
Port 9	43	P90	General-purpose output port	General-purpose input port	EXMOD0	Extension mode setting input 0
	42	P91	General-purpose output port	General-purpose input port	EXMOD1	Extension mode setting input 1
	40	P92	General-purpose output port		$\overline{\text{RE}}$	Memory read signal output
	39	P93	General-purpose output port		$\overline{\text{WE0}}$	Memory write signal output
	38	P94	General-purpose output port		$\overline{\text{WE1}}$	Memory write signal output
	37	P95	General-purpose output port	General-purpose input port	$\overline{\text{DK}}$	Data acknowledge signal input
	36	P96	General-purpose output port	General-purpose input port	$\overline{\text{BR}}$	Bus authority request signal input
	34	P97	General-purpose output port		SYSCLK	System clock output

[Note 1]

▨ : When reset (whether in address/data separate mode or address/data multiplex mode)

[Note 2]

When the bus authority is granted, $\overline{\text{WE1}}$, $\overline{\text{WE0}}$, and $\overline{\text{RE}}$ go to high impedance.

15.12 Port A

15.12.1 Block Diagram

Fig. 15-12-1 shows a block diagram for port A.

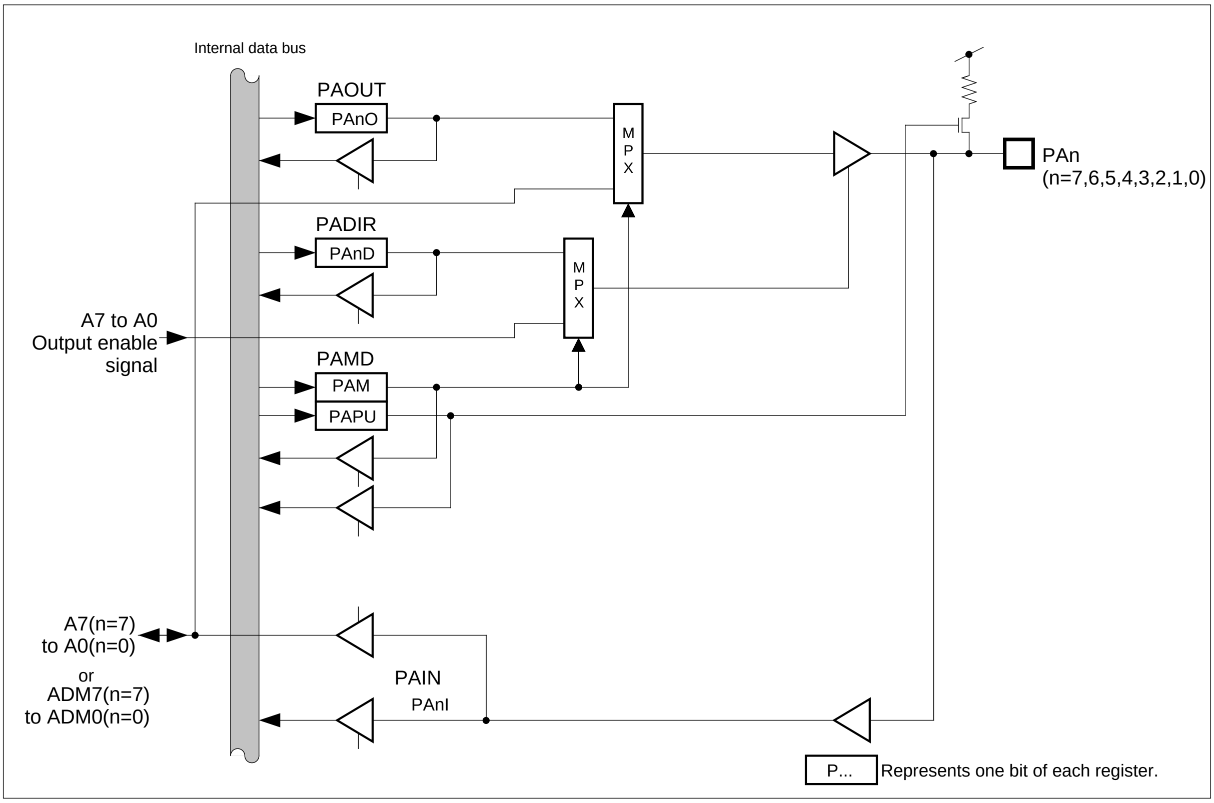


Fig. 15-12-1 Port A Block Diagram (PA7 to PA0)

15.12.2 Register Descriptions

Port A is a general-purpose input/output port that is also used for address bus signals A[7:0], and address/ data signals ADM[7:0].

Each register for port A is described below.

Port A output register

Register symbol: PAOUT
Address: x'36008014
Purpose: This register sets the data to be output on port A.

Bit No.	7	6	5	4	3	2	1	0
Bit name	PA7O	PA6O	PA5O	PA4O	PA3O	PA2O	PA1O	PA0O
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Port A pin register

Register symbol: PAIN
Address: x'36008094
Purpose: This register reads the value of the port A pins.

Bit No.	7	6	5	4	3	2	1	0
Bit name	PA7I	PA6I	PA5I	PA4I	PA3I	PA2I	PA1I	PA0I
Reset	X	X	X	X	X	X	X	X
Access	R	R	R	R	R	R	R	R

Port A input/output control register

Register symbol: PADIR
Address: x'36008074
Purpose: This register sets the port A pins for input or output.
Valid when the PAM is "1".
(0: input; 1: output)

Bit No.	7	6	5	4	3	2	1	0
Bit name	PA7D	PA6D	PA5D	PA4D	PA3D	PA2D	PA1D	PA0D
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Port A output mode register

Register symbol: PAMD
Address: x'36008034
Purpose: When PAM is "1", the port A pins are set as a general-purpose port; when PAM is "0", the port A pins are set as address pins.
When PAPU is "1", the port A pins are pulled up.
Note : The setting of PAPU that pulls up (or does not pull up) the port A pins can be made regardless of the value of PAM.

Bit No.	7	6	5	4	3	2	1	0
Bit name	-	-	-	-	-	-	PAPU	PAM
Reset	0	0	0	0	0	0	0(1)	0
Access	R	R	R	R	R	R	R/W	R/W

() are set in addresses/data multiplex mode.

15.12.3 Pin Configurations

Table 15-12-1 shows the pin configurations for port A.

Table 15-12-1 Port A Configuration

Port	Pin No.	PAn	PAM = "1"		PAM = "0"	
			PAnD = "1"	PAnD = "0"		
Port A	24	PA0	General-purpose output port	General-purpose input port	A0 <<ADM0>> *1	Address output <<Address/data input/output>>
	23	PA1	General-purpose output port	General-purpose input port	A1 <<ADM1>> *1	Address output <<Address/data input/output>>
	22	PA2	General-purpose output port	General-purpose input port	A2 <<ADM2>> *1	Address output <<Address/data input/output>>
	20	PA3	General-purpose output port	General-purpose input port	A3 <<ADM3>> *1	Address output <<Address/data input/output>>
	19	PA4	General-purpose output port	General-purpose input port	A4 <<ADM4>> *1	Address output <<Address/data input/output>>
	18	PA5	General-purpose output port	General-purpose input port	A5 <<ADM5>> *1	Address output <<Address/data input/output>>
	17	PA6	General-purpose output port	General-purpose input port	A6 <<ADM6>> *1	Address output <<Address/data input/output>>
	16	PA7	General-purpose output port	General-purpose input port	A7 <<ADM7>> *1	Address output <<Address/data input/output>>

[Note 1]

▨ : When reset (whether in address/data separate mode or address/data multiplex mode)

*1 : In the event of a reset in address/data multiplex mode, the PAPU bit in the PAMD register is set to "1" and the address/data pins (the 8 bits ADM[7:0]) are pulled up.

<<>> : These pins are set in address/data multiplex mode.

[Note 2]

When the bus authority is granted, A7 to A0 (ADM7 to ADM0) go to high impedance.

15.13 Port B

15.13.1 Block Diagram

Fig. 15-13-1 shows a block diagram for port B.

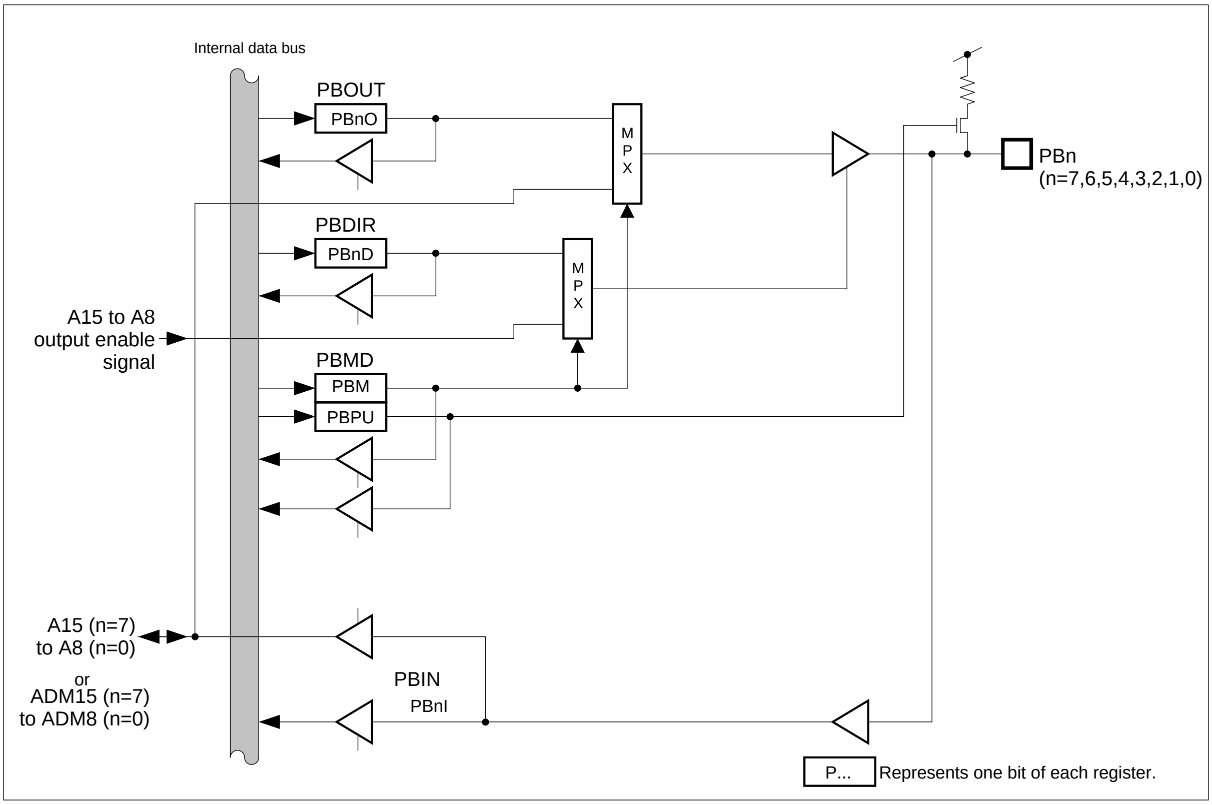


Fig. 15-13-1 Port B Block Diagram (PB7 to PB0)

15.13.2 Register Descriptions

Port B is a general-purpose input/output port that is also used for address bus signals A[15:8], and address/ data signals ADM[15:8].

Each register for port B is described below.

Port B output register

Register symbol: PBOUT
Address: x'36008015
Purpose: This register sets the data to be output on port B.

Bit No.	7	6	5	4	3	2	1	0
Bit name	PB7O	PB6O	PB5O	PB4O	PB3O	PB2O	PB1O	PB0O
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Port B pin register

Register symbol: PBIN
Address: x'36008095
Purpose: This register reads the value of the port B pins.

Bit No.	7	6	5	4	3	2	1	0
Bit name	PB7I	PB6I	PB5I	PB4I	PB3I	PB2I	PB1I	PB0I
Reset	X	X	X	X	X	X	X	X
Access	R	R	R	R	R	R	R	R

Port B input/output control register

Register symbol: PBDIR
Address: x'36008075
Purpose: This register sets the port B pins for input or output.
Valid when PBM is "1".
(0: input; 1: output)

Bit No.	7	6	5	4	3	2	1	0
Bit name	PB7D	PB6D	PB5D	PB4D	PB3D	PB2D	PB1D	PB0D
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Port B output mode register

Register symbol: PBMD
Address: x'36008035
Purpose: When PBM is "1", the port B pins are set as a general-purpose port; when PBM is "0", the port B pins are set as address pins.
When PBPU is "1", the port B pins are pulled up.
Note : The setting of PBPU that pulls up (or does not pull up) the port B pins can be made regardless of the value of PBM.

Bit No.	7	6	5	4	3	2	1	0
Bit name	-	-	-	-	-	-	PBPU	PBM
Reset	0	0	0	0	0	0	0(1)	0
Access	R	R	R	R	R	R	R/W	R/W

() are set in address/data multiplex mode.

15.13.3 Pin Configurations

Table 15-13-1 shows the pin configurations for port B.

Table 15-13-1 Port B Configuration

Port	Pin	PBn	PBM = "1"		PBM = "0"	
	No.		PBnD = "1"	PBnD = "0"		
Port B	14	PB0	General-purpose output port	General-purpose input port	A8 <ADM8>> *1	Address output <<Address/data input/output>>
	13	PB1	General-purpose output port	General-purpose input port	A9 <<ADM9>> *1	Address output <<Address/data input/output>>
	12	PB2	General-purpose output port	General-purpose input port	A10 <<ADM10>> *1	Address output <<Address/data input/output>>
	11	PB3	General-purpose output port	General-purpose input port	A11 <<ADM11>> *1	Address output <<Address/data input/output>>
	10	PB4	General-purpose output port	General-purpose input port	A12 <<ADM12>> *1	Address output <<Address/data input/output>>
	8	PB5	General-purpose output port	General-purpose input port	A13 <<ADM13>> *1	Address output <<Address/data input/output>>
	7	PB6	General-purpose output port	General-purpose input port	A14 <<ADM14>> *1	Address output <<Address/data input/output>>
	6	PB7	General-purpose output port	General-purpose input port	A15 <<ADM15>> *1	Address output <<Address/data input/output>>

[Note 1]

▨ : When reset (whether in address/data separate mode or address/data multiplex mode)

*1 : In the event of a reset in address/data multiplex mode, the PBPU bit in the PBMD register is set to "1" and the address/data pins (the 8 bits ADM[15:8]) are pulled up.

<<>> : These pins are set in address/data multiplex mode.

[Note 2]

When the bus authority is granted, A15 to A8 (ADM15 to ADM8) go to high impedance.

15.14 Port C

15.14.1 Block Diagram

Fig. 15-14-1 shows a block diagram for port C.

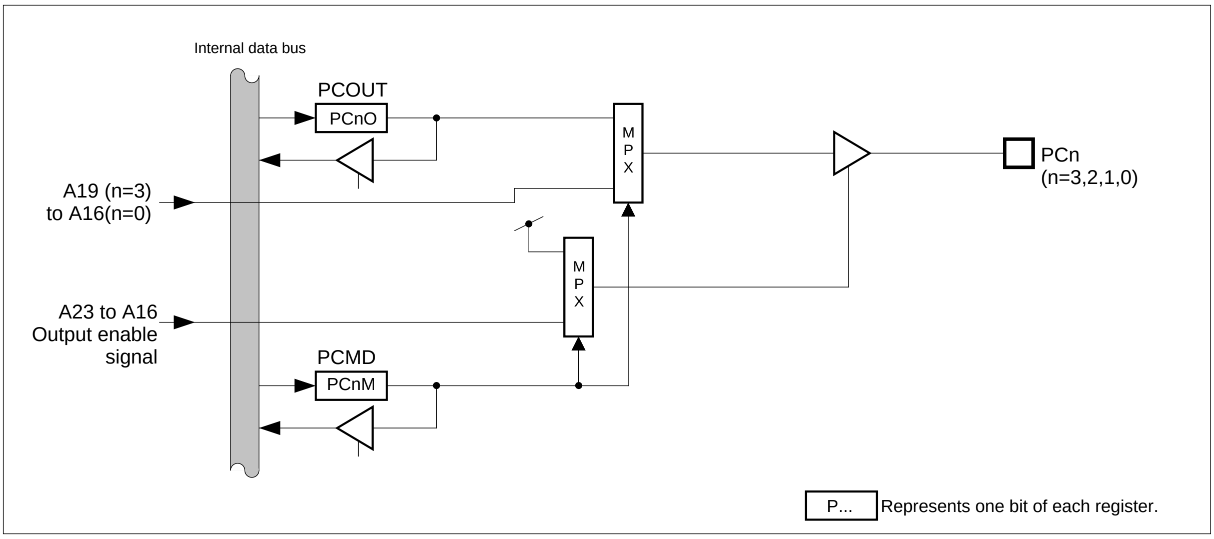


Fig. 15-14-1 Port C Block Diagram (PC3 to PC0)

15.14.2 Register Descriptions

Port C is a general-purpose output port that is also used for address bus signals A[19:16].

Each register for port C is described below.

Port C output register

Register symbol: PCOUT
Address: x'36008018
Purpose: This register sets the data to be output on port C.

Bit No.	7	6	5	4	3	2	1	0
Bit name	-	-	-	-	PC3O	PC2O	PC1O	PC0O
Reset	0	0	0	0	0	0	0	0
Access	R	R	R	R	R/W	R/W	R/W	R/W

Port C output mode register

Register symbol: PCMD
Address: x'36008038
Purpose: This register selects the content output on the port C pins.

Bit No.	7	6	5	4	3	2	1	0
Bit name	-	-	-	-	PC3M	PC2M	PC1M	PC0M
Reset	0	0	0	0	0	0	0	0
Access	R	R	R	R	R/W	R/W	R/W	R/W

PC3M 0: Address output (A19)
1: General-purpose output port (PC3)

PC2M 0: Address output (A18)
1: General-purpose output port (PC2)

PC1M 0: Address output (A17)
1: General-purpose output port (PC1)

PC0M 0: Address output (A16)
1: General-purpose output port (PC0)

15.14.3 Pin Configurations

Table 15-14-1 shows the pin configurations for port C.

Table 15-14-1 Port C Configuration

Port	Pin No.	PCn	PCnM = "1"	PCnM = "0"	
Port C	5	PC0	General-purpose output port	A16	Address output
	4	PC1	General-purpose output port	A17	Address output
	2	PC2	General-purpose output port	A18	Address output
	1	PC3	General-purpose output port	A19	Address output

[Note 1]

▨ : When reset (whether in address/data separate mode or address/data multiplex mode)

[Note 2]

When the bus authority is granted, A19 to A16 go to high impedance.

15.15 Treatment of Unused Pins

Unused pins should be treated as shown in Table 15-15-1 below.

Table 15-15-1 Treatment of Unused Pins

Pin name	Treatment
PC3/A19, PC2/A18, PC1/A17, PC0/A16	Set as port and leave open.
PB7/ADM15/A15, PB6/ADM14/A14, PB5/ADM13/A13, PB4/ADM12/A12, PB3/ADM11/A11, PB2/ADM10/A10, PB1/ADM9/A9, PB0/ADM8/A8, PA7/ADM7/A7, PA6/ADM6/A6, PA5/ADM5/A5, PA4/ADM4/A4, PA3/ADM3/A3, PA2/ADM2/A2, PA1/ADM1/A1, PA0/ADM0/A0	Either set as input port and connect to VDD or VSS via individual resistors (or else use the built-in pull-up resistance by setting the register appropriately), or set as output port and leave open.
OSCO	Leave open.
SYSCLK/P97	Set as port and leave open.
P96/$\overline{\text{BR}}$, P95/$\overline{\text{DK}}$	Either set as input port and connect to VDD or VSS via individual resistors, or set as output port and leave open.
P94/$\overline{\text{WE1}}$, P93/$\overline{\text{WE0}}$, P92/$\overline{\text{RE}}$	Set as port and leave open.
P83/AN3/IRQ7, P82/AN2/IRQ6, P81/AN1/IRQ5, P80/AN0/IRQ4	Connect to AVDD or VSS via individual resistors.
VREFH	Connect to AVDD.
P73/A23/$\overline{\text{CS3}}$, P72/$\overline{\text{RAS2}}$/$\overline{\text{CS2}}$, P71/$\overline{\text{RAS1}}$/$\overline{\text{CS1}}$, P70/$\overline{\text{CS0}}$	Set as port and leave open.
P63/IRQ3/$\overline{\text{ADTRG}}$/TM10IOB, P62/IRQ2/TM10IOA, P61/IRQ1/TM7IO, P60/IRQ0/TM6IO	Either set as input port and connect to VDD or VSS via individual resistors, or set as output port and leave open.
$\overline{\text{NMIRQ}}$	Connect to VDD via a resistor.
P55/SBO3/TM5IO/TM13IO, P54/SBI3/TM4IO/TM12IO, P53/SBT3/TM3IO/TM11IO, P52/SBO2/TM2IO, P51/SBI2/TM1IO, P50/SBT2/TM0IO, P45/SBO1/$\overline{\text{DWE}}$, P44/SBI1/$\overline{\text{DCAS1}}$, P43/SBT1/$\overline{\text{DCAS0}}$, P42/SBO0, P41/SBI0, P40/SBT0, P30/$\overline{\text{BG}}$	Either set as input port and connect to VDD or VSS via individual resistors, or set as output port and leave open.
P27/D15, P26/D14, P25/D13, P24/D12, P23/D11, P22/D10, P21/D9, P20/D8, P17/D7, P16/D6, P15/D5, P14/D4, P13/D3, P12/D2, P11/RWSEL/D1, P10/AS/D0	Either set as input port and connect to VDD or VSS via individual resistors (or else use the built-in pull-up resistance by setting the register appropriately), or set as output port and leave open.
P02/$\overline{\text{CAS}}$/A22, P01/A21, P00/A20	Set as port and leave open.

16. Internal Flash Memory

16.1 Overview

The MN1030F01K has 256 KB of internal flash memory for use as instruction memory in place of instruction ROM. Using flash memory makes it easy to make changes to a stored program, which makes it possible to reduce program development time and permits the creation of a highly flexible system.

16.2 Features

The features of the internal flash memory are described below.

- Capacity: 256 KB
- Permits batch erasure of entire area and erasure of individual 8 KB blocks.
- 8-byte page program permits fast data writing.
- Supports two flash memory overwrite modes (flash memory mode and on-board write mode).
- On-board write mode includes mechanisms designed to prevent accidental erasure or writing of flash memory.

16.3 Block Diagram

Fig. 16-3-1 shows the block diagram of flash memory and related blocks.

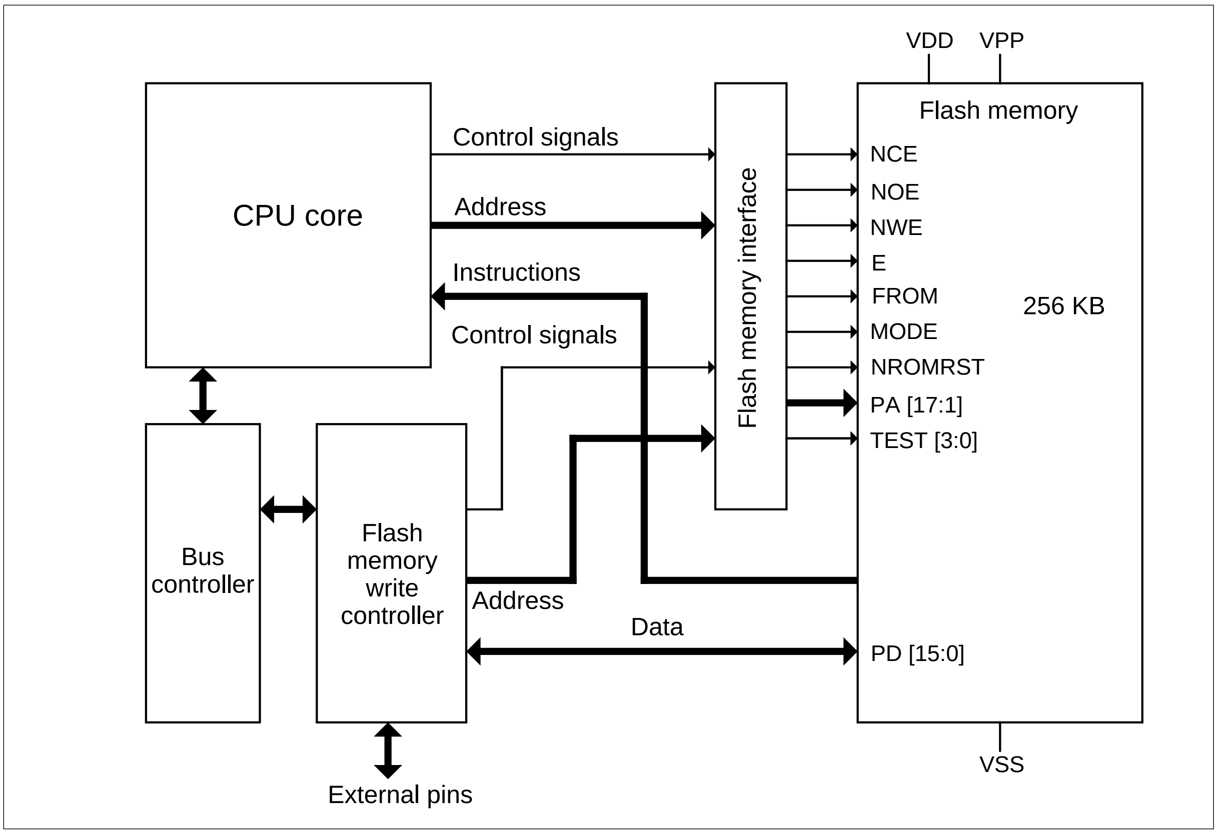


Fig. 16-3-1 Flash Memory Block Diagram

16.4 Flash Memory Overwrite Mode and Settings

There are two flash memory overwrite modes: flash memory mode and on-board write mode. Table 16-4-1 lists the mode settings through the external pins.

Flash memory mode is used to overwrite the internal flash memory with a ROM writer. In this mode, the flash memory inputs and outputs are connected to external pins.

On-board write mode is used to overwrite the internal flash memory via software. This makes it possible to overwrite the internal flash memory while the microcontroller is still mounted on a board. The address/data separate mode or address/data multiplex mode may apply as the on-board write mode.

Table 16-4-1 Mode Settings through the External Pins

Mode name	MMOD1	MMOD0	EXMOD1	EXMOD0
Flash memory mode	H	H	L	H
On-board write mode (memory extension mode 16-bit data address/data separate)	L	L	L	H
On-board write mode (memory extension mode 16-bit data address/data multiplex)	L	L	H	H

16.5 Flash Memory Mode

16.5.1 Description of External Pins

Fig. 16-5-1 and Table 16-5-1 show the pin assignments for the MN1030F01K in flash memory mode.

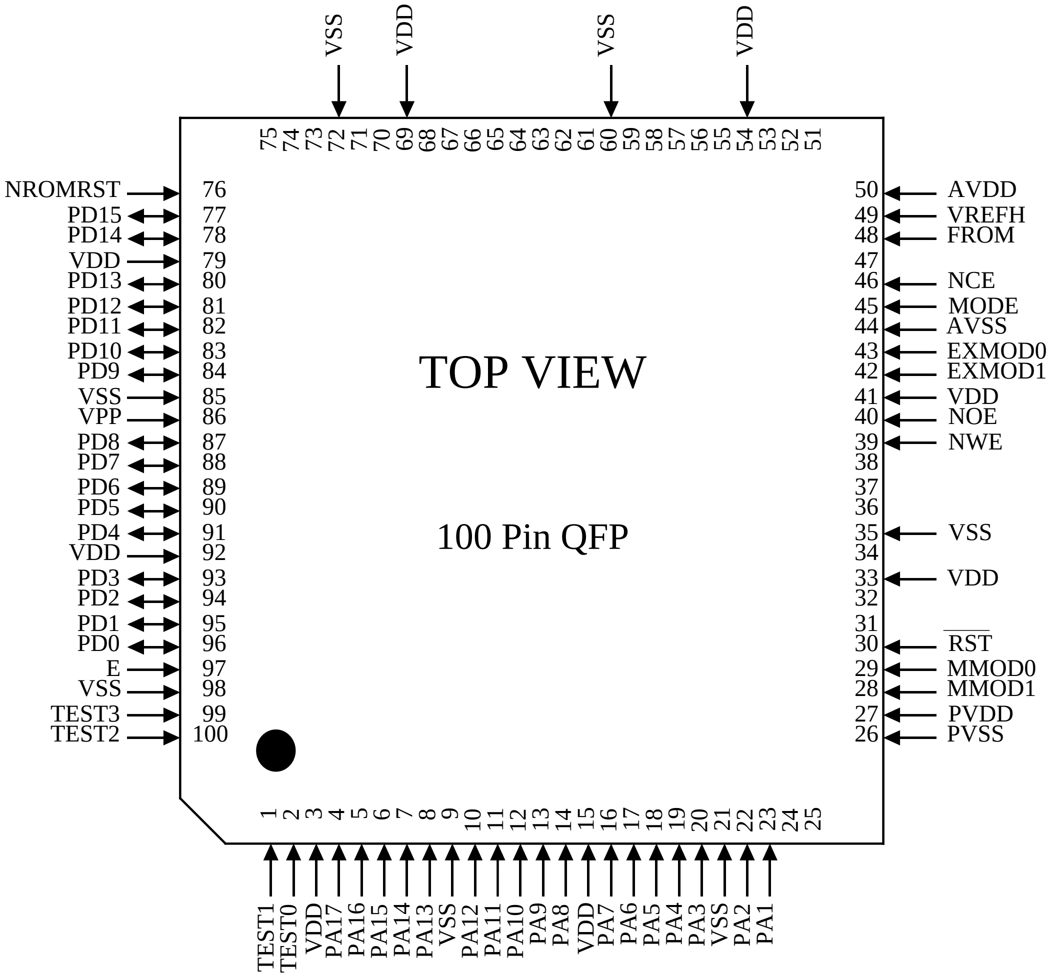


Fig. 16-5-1 MN1030F01K Pin Assignments in Flash Memory Mode

Table 16-5-1 MN1030F01K Pin Assignments

Pin No.	Pin Name	I/O	Pin No.	Pin Name	I/O	Pin No.	Pin Name	I/O	Pin No.	Pin Name	I/O
1	TEST1	L	26	PVSS		51	—	O	76	NROMRST	I
2	TEST0	L	27	PVDD		52	—	O	77	PD15	I/O
3	VDD		28	MMOD1	H	53	—	O	78	PD14	I/O
4	PA17	I	29	MMOD0	H	54	VDD		79	VDD	
5	PA16	I	30	$\overline{\text{RST}}$	L	55	—	O	80	PD13	I/O
6	PA15	I	31	—	O	56	—	O	81	PD12	I/O
7	PA14	I	32	—	H	57	—	O	82	PD11	I/O
8	PA13	I	33	VDD		58	—	O	83	PD10	I/O
9	VSS		34	—	O	59	—	O	84	PD9	I/O
10	PA12	I	35	VSS		60	VSS		85	VSS	
11	PA11	I	36	—	H	61	—	O	86	VPP	
12	PA10	I	37	—	H	62	—	O	87	PD8	I/O
13	PA9	I	38	—	H	63	—	O	88	PD7	I/O
14	PA8	I	39	NWE	I	64	—	O	89	PD6	I/O
15	VDD		40	NOE	I	65	—	O	90	PD5	I/O
16	PA7	I	41	VDD		66	—	O	91	PD4	I/O
17	PA6	I	42	EXMOD1	L	67	—	O	92	VDD	
18	PA5	I	43	EXMOD0	H	68	—	O	93	PD3	I/O
19	PA4	I	44	AVSS		69	VDD		94	PD2	I/O
20	PA3	I	45	MODE	I	70	—	O	95	PD1	I/O
21	VSS		46	NCE	I	71	—	O	96	PD0	I/O
22	PA2	I	47	—	H	72	VSS		97	E	I
23	PA1	I	48	FROM	I	73	—	O	98	VSS	
24	—	I	49	VREFH		74	—	O	99	TEST3	L
25	—	I	50	AVDD		75	—	O	100	TEST2	L

I: Input; O: Output; I/O: Input/output; H: High level input; L: Low level input

Table 16-5-2 lists the functions of the external pins in flash memory mode.

Table 16-5-2 Pin Functions

Pin Name	Input/Output	Description
PA[17:1]	Input	Address
PD[15:0]	Input/Output	Data
NCE	Input	Chip enable
MODE	Input	Mode
NOE	Input	Output enable
NWE	Input	Write enable
E	Input	Erase enable
FROM	Input	Flash memory/microcontroller mode switch
TEST[3:0]	Input	Test signal * Input x'0.
NROMRST	Input	Reset
VDD		Power supply
VPP		Power supply
VSS		Power supply

When first applying power, it is necessary to input a signal that is low for at least 1 ms to the reset pin NROMRST.

16.5.2 Erasure Blocks

The flash memory is partitioned into 32 8 KB erasure blocks. Fig. 16-5-2 shows the configuration of the flash memory erasure blocks and their correspondence with each of the bits in the erasure block registers that are used to specify which blocks to erase. After setting the erasure block registers in erasure block setting mode, block erase mode is used to erase the blocks that correspond to the bits for which "1" was specified in the erasure block registers.

In all erase mode, all 256 KB memory cells are erased in a single operation.

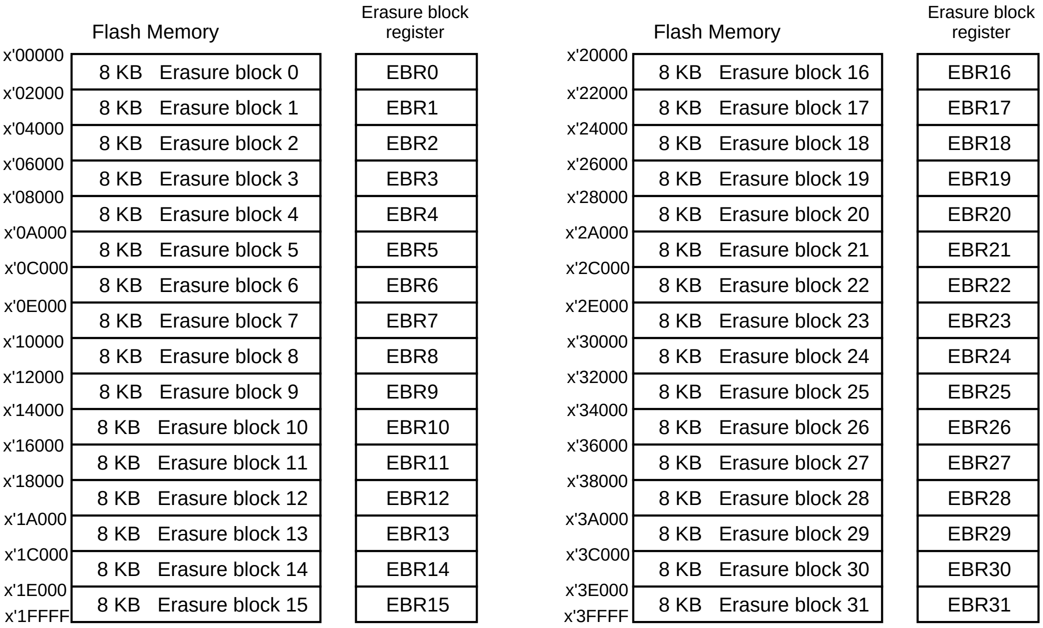


Fig. 16-5-2 Flash Memory Erasure Blocks

16.6 On-board Write Mode

In on-board write mode, flash memory is overwritten by manipulating the control registers through software.

Table 16-6-1 lists the control registers to be used in on-board write mode.

Table 16-6-1 Flash Memory Register List

Address	Registername	Symbol	Number of bits	Initial value	Access size
x'34010000	Flash on-board rewrite control register	FCREG	16	x'0017	8, 16
x'34010004	Flash data register	FDREG	16	x'0000	8, 16
x'34010008	Flash address register (Lower)	FAREG	16	x'0000	8, 16
x'3401000C	Flash address register (Upper)	FAREGEX	16	x'0000	8, 16
x'34010010	Flash on-board rewrite enable register	FBEWER	16	x'0000	8, 16
x'34010014	Flash memory mode register	FLMODR	8	*1	8

*1: FLMODR[3:0] uses the values of MMOD1 and 0 and EXMOD1 and 0, and FLMODR[7:4] is x'0.

If an on-board write is performed, some bus control signal pins (except for $\overline{\text{RE}}$, $\overline{\text{WE1}}$, and $\overline{\text{WE0}}$) and the address and data signal pins operate.
The pins that operate are listed below:

Address/data multiplex	AS RWSEL ADM [15:0]
Address/data separate	A [12:0] D [15:0]

The $\overline{\text{RE}}$, $\overline{\text{WE1}}$, and $\overline{\text{WE0}}$ signal pins each output a high signal.
If the pins are set for other signals by the I/O port register settings, they operate in accordance with the register settings.

Design the board of any external device that is to be connected to above signal pins in such a way that no difficulty will be encountered even if the above signal pins do operate.

17. Ordering Mask ROM

17.1 Overview

This chapter describes the procedure for ordering mask ROM. This chapter also describes the difference in programming when using a product that has on-chip flash memory versus a mask product, and explains how to order ROM, etc.

17.2 Procedure for Ordering ROM

When program development with a product that has on-chip flash memory has been conducted using a flash overwrite program (loader program), process the flash memory program by either of the following methods when ordering the mask ROM product.

[Ordering method 1]
Refer to Fig. 17-2-1.
Delete the loader program portion (8 KB), and then recompile so that the start of the user program is located at x’40000000. (The address x’40000008 should be used as the starting address for the user non-maskable interrupt processing routine. This is not necessary if non-maskable interrupts are not being used, however.)
When using this method, however, the program must be relocatable.

[Ordering method 2]
Refer to Fig. 17-2-2.
Rewrite the loader program so that the user program executes without referencing the flash memory mode register (FLMODR).
Because the execution address after a reset state is released is x’40000000, and the execution address when a non-maskable interrupt is generated is x’40000008, place an instruction that branches to the start of the user program in address x’40000000, and place an instruction that branches to the start of the user non-maskable interrupt processing routine in address x’40000008.
If the user is not using non-maskable interrupt processing, the branch instruction in address x’40000008 is not needed.

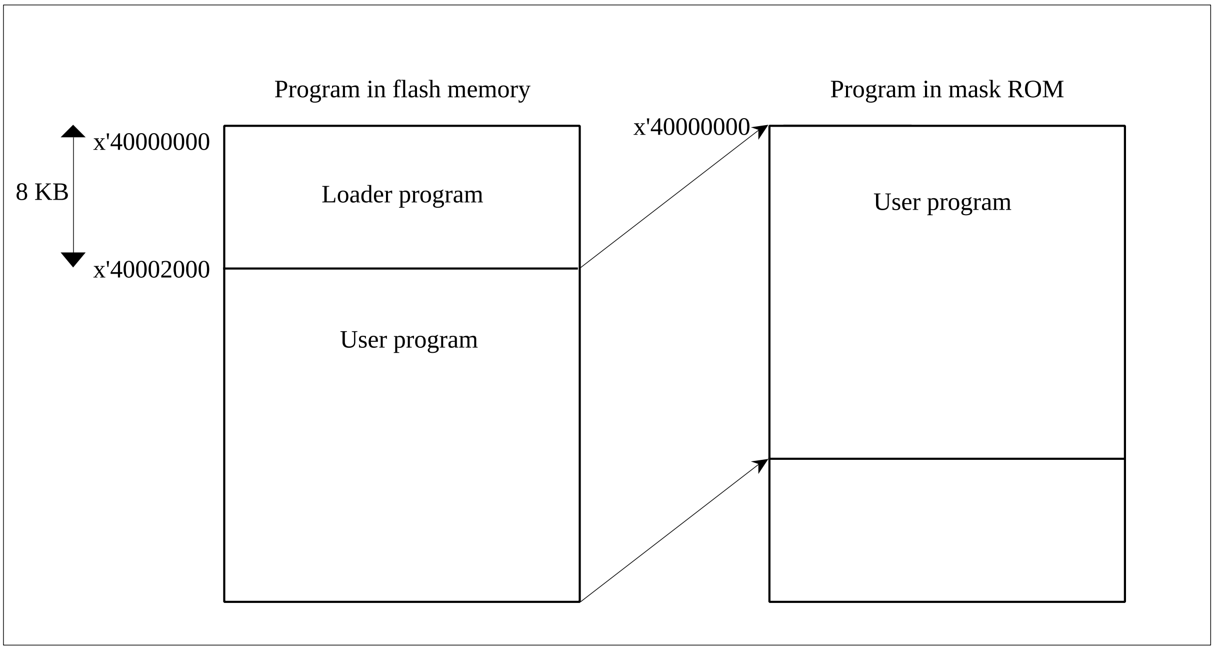


Fig. 17-2-1 ROM Ordering Method 1

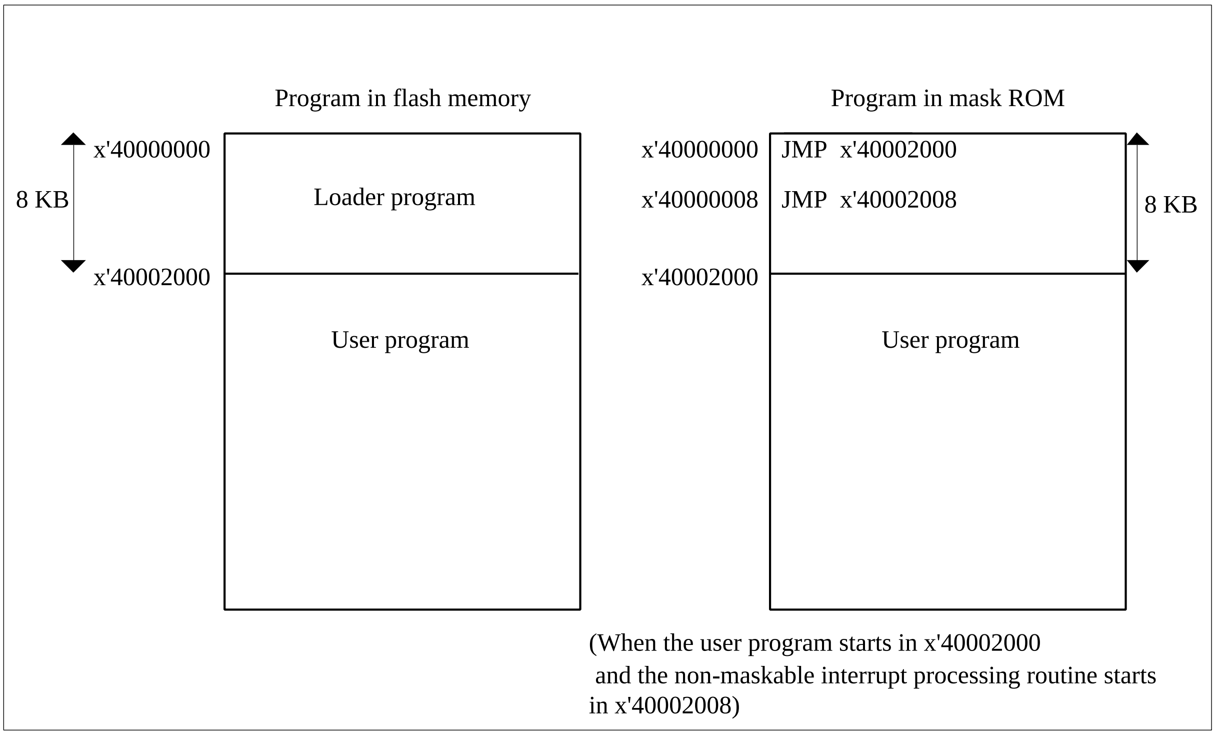


Fig. 17-2-2 ROM Ordering Method 2

Appendix

Appendix A. Register Map List

Address	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	
x'2000000X			IVAR3				IVAR2				IVAR1				IVAR0		CPU
x'2000001X							IVAR6				IVAR5				IVAR4		
x'2000002X															MEMCTRC		
x'2000004X															CPUM		
x'3200002X									MEMCTR3B		MEMCTR2B		MEMCTR1B		MEMCTR0B		Memory control
x'3200003X									MEMCTR3A		MEMCTR2A		MEMCTR1A		MEMCTR0A		
x'3200004X											PRAR		REFCNT		DRAMCTR		
x'3200400X															CKCTR		
x'3400010X			G3ICR				G2ICR								NMICR		Interrupts
x'3400011X			G7ICR				G6ICR				G5ICR				G4ICR		
x'3400012X			G11ICR				G10ICR				G9ICR				G8ICR		
x'3400013X			G15ICR				G14ICR				G13ICR				G12ICR		
x'3400014X			G19ICR				G18ICR				G17ICR				G16ICR		
x'3400020X															IAGR		
x'3400028X															EXTMD		
x'3400040X															ADCTR		A/D
x'3400041X			AD3BUF				AD2BUF				AD1BUF				AD0BUF		

Note: Accessing areas that are not mounted is prohibited. Operation is not guaranteed if an area that is not mounted is accessed.

Address	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	
x'3400080X			SC0STR				SC0 RXB	SC0 TXB				SC0 ICR			SC0CTR		Serial interface
x'3400081X				SC1 STR			SC1 RXB	SC1 TXB				SC1 ICR			SC1CTR		
x'3400082X				SC2 STR			SC2 RXB	SC2 TXB				SC2 ICR			SC2CTR		
x'3400083X			SC3 TIM	SC3 STR			SC3 RXB	SC3 TXB				SC3 ICR			SC3CTR		
x'3400100X					TMB MD	TMA MD	TM9 MD	TM8 MD	TM7 MD	TM6 MD	TM5 MD	TM4 MD	TM3 MD	TM2 MD	TM1 MD	TM0 MD	8-bit timer
x'3400101X					TMB BR	TMA BR	TM9 BR	TM8 BR	TM7 BR	TM6 BR	TM5 BR	TM4 BR	TM3 BR	TM2 BR	TM1 BR	TM0 BR	
x'3400102X					TMB BC	TMA BC	TM9 BC	TM8 BC	TM7 BC	TM6 BC	TM5 BC	TM4 BC	TM3 BC	TM2 BC	TM1 BC	TM0 BC	
x'3400103X					TMB CMP	TMA CMP	TM9 CMP	TM8 CMP	TM7 CMP	TM6 CMP	TM5 CMP	TM4 CMP					
x'3400107X															TMPS CNT	TMOSL	
x'3400108X										TM13 MD		TM12 MD		TM11 MD	TM10MD		16-bit timer
x'3400109X									TM13BR		TM12BR		TM11BR				
x'340010AX									TM13BC		TM12BC		TM11BC		TM10BC		
x'340010BX															TM10 MDB	TM10 MDA	
x'340010CX															TM10CA		
x'340010DX															TM10CB		
x'3400400X								WDCTR				RST CTR				WDBC	Watchdog timer
x'3401000X			FAREGEX				FAREG				FDREG				FCREG		Flash memory control
x'3401001X												FL MODR			FBEWER		

▨ :This register is installed only for versions with a flash memory.

Note: Accessing areas that are not mounted is prohibited. Operation is not guaranteed if an area that is not mounted is accessed.

Address	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	
x'3600800X			P7OUT	P6OUT			P5OUT	P4OUT			P3OUT	P2OUT			P1OUT	P0OUT	I/O port
x'3600801X								PCOUT			PBOUT	PAOUT			P9OUT		
x'3600802X			P7MD	P6MD			P5MD	P4MD			P3MD	P2MD			P1MD	P0MD	
x'3600803X								PCMD			PBMD	PAMD			P9MD	P8AD	
x'3600804X			P7SS				P5SS	P4SS								P0SS	
x'3600805X																	
x'3600806X				P6DIR			P5DIR	P4DIR			P3DIR	P2DIR			P1DIR		
x'3600807X											PBDIR	PADIR			P9DIR		
x'3600808X				P6IN			P5IN	P4IN			P3IN	P2IN			P1IN		
x'3600809X											PBIN	PAIN			P9IN	P8IN	

Note: Accessing areas that are not mounted is prohibited. Operation is not guaranteed if an area that is not mounted is accessed.

Appendix B. Instruction Set

List of Instructions (Code Length, Execution Cycle*)

Instruction		Source	Destination	Format	Code length	Execution Cycle	Remarks
MOV	MOV	Dm	Dn	S0	1	1	
	MOV	Dm	An	D0	2	1	
	MOV	Am	Dn	D0	2	1	
	MOV	Am	An	S0	1	1	
	MOV	SP	An	S0	1	1	
	MOV	Am	SP	D0	2	1	
	MOV	PSW	Dn	D0	2	1	
	MOV	Dm	PSW	D0	2	1	
	MOV	MDR	Dn	D0	2	1	
	MOV	Dm	MDR	D0	2	1	
	MOV	(Am)	Dn	S0	1	1	
	MOV	(d8,Am)	Dn	D1	3	1	
	MOV	(d16,Am)	Dn	D2	4	1	
	MOV	(d32,Am)	Dn	D4	6	2	
	MOV	(d8,SP)	Dn	S1	2	1	
	MOV	(d16,SP)	Dn	D2	4	1	
	MOV	(d32,SP)	Dn	D4	6	2	
	MOV	(Di,Am)	Dn	D0	2	1	
	MOV	(abs16)	Dn	S2	3	1	
	MOV	(abs32)	Dn	D4	6	2	
	MOV	(Am)	An	D0	2	1	
	MOV	(d8,Am)	An	D1	3	1	
	MOV	(d16,Am)	An	D2	4	1	
	MOV	(d32,Am)	An	D4	6	2	
	MOV	(d8,SP)	An	S1	2	1	
	MOV	(d16,SP)	An	D2	4	1	
	MOV	(d32,SP)	An	D4	6	2	
	MOV	(Di,Am)	An	D0	2	1	
	MOV	(abs16)	An	D2	4	1	
	MOV	(abs32)	An	D4	6	2	
	MOV	(d8,Am)	SP	D1	3	1	
	MOV	Dm	(An)	S0	1	1	
	MOV	Dm	(d8,An)	D1	3	1	
	MOV	Dm	(d16,An)	D2	4	1	
	MOV	Dm	(d32,An)	D4	6	2	
	MOV	Dm	(d8,SP)	S1	2	1	
	MOV	Dm	(d16,SP)	D2	4	1	
	MOV	Dm	(d32,SP)	D4	6	2	
	MOV	Dm	(Di,An)	D0	2	2	
	MOV	Dm	(abs16)	S2	3	1	
	MOV	Dm	(abs32)	D4	6	2	
	MOV	Am	(An)	D0	2	1	
	MOV	Am	(d8,An)	D1	3	1	
	MOV	Am	(d16,An)	D2	4	1	
	MOV	Am	(d32,An)	D4	6	2	
	MOV	Am	(d8,SP)	S1	2	1	
	MOV	Am	(d16,SP)	D2	4	1	
	MOV	Am	(d32,SP)	D4	6	2	
	MOV	Am	(Di,An)	D0	2	2	
	MOV	Am	(abs16)	D2	4	1	
	MOV	Am	(abs32)	D4	6	2	
	MOV	SP	(d8,An)	D1	3	1	
	MOV	imm8	Dn	S1	2	1	
	MOV	imm16	Dn	S2	3	1	
	MOV	imm32	Dn	D4	6	2	
	MOV	imm8	An	S1	2	1	
	MOV	imm16	An	S2	3	1	
	MOV	imm32	An	D4	6	2	

Execution cycle is defined under the following conditions:

(1) No pipeline hazard

(2) 2-cycle of instruction fetch, 1-cycle of data load/store

Instruction		Source	Destination	Format	Code length	Execution Cycle	Remarks
MOVBU	MOVBU	(Am)	Dn	D0	2	1	
	MOVBU	(d8,Am)	Dn	D1	3	1	
	MOVBU	(d16,Am)	Dn	D2	4	1	
	MOVBU	(d32,Am)	Dn	D4	6	2	
	MOVBU	(d8,SP)	Dn	D1	3	1	
	MOVBU	(d16,SP)	Dn	D2	4	1	
	MOVBU	(d32,SP)	Dn	D4	6	2	
	MOVBU	(Di,Am)	Dn	D0	2	1	
	MOVBU	(abs16)	Dn	S2	3	1	
	MOVBU	(abs32)	Dn	D4	6	2	
	MOVBU	Dm	(An)	D0	2	1	
	MOVBU	Dm	(d8,An)	D1	3	1	
	MOVBU	Dm	(d16,An)	D2	4	1	
	MOVBU	Dm	(d32,An)	D4	6	2	
	MOVBU	Dm	(d8,SP)	D1	3	1	
	MOVBU	Dm	(d16,SP)	D2	4	1	
	MOVBU	Dm	(d32,SP)	D4	6	2	
	MOVBU	Dm	(Di,An)	D0	2	2	
	MOVBU	Dm	(abs16)	S2	3	1	
	MOVBU	Dm	(abs32)	D4	6	2	
MOVHU	MOVHU	(Am)	Dn	D0	2	1	
	MOVHU	(d8,Am)	Dn	D1	3	1	
	MOVHU	(d16,Am)	Dn	D2	4	1	
	MOVHU	(d32,Am)	Dn	D4	6	2	
	MOVHU	(d8,SP)	Dn	D1	3	1	
	MOVHU	(d16,SP)	Dn	D2	4	1	
	MOVHU	(d32,SP)	Dn	D4	6	2	
	MOVHU	(Di,Am)	Dn	D0	2	1	
	MOVHU	(abs16)	Dn	S2	3	1	
	MOVHU	(abs32)	Dn	D4	6	2	
	MOVHU	Dm	(An)	D0	2	1	
	MOVHU	Dm	(d8,An)	D1	3	1	
	MOVHU	Dm	(d16,An)	D2	4	1	
	MOVHU	Dm	(d32,An)	D4	6	2	
	MOVHU	Dm	(d8,SP)	D1	3	1	
	MOVHU	Dm	(d16,SP)	D2	4	1	
	MOVHU	Dm	(d32,SP)	D4	6	2	
	MOVHU	Dm	(Di,An)	D0	2	2	
	MOVHU	Dm	(abs16)	S2	3	1	
	MOVHU	Dm	(abs32)	D4	6	2	
EXT	EXT		Dn	D0	2	1	
EXTB	EXTB		Dn	S0	1	1	
EXTBU	EXTBU		Dn	S0	1	1	
EXTH	EXTH		Dn	S0	1	1	
EXTHU	EXTHU		Dn	S0	1	1	
MOVM	MOVM	(SP)	regs	S1	2	1	Registers specified by regs = 0
						2	Registers specified by regs = 1
						3	Registers specified by regs = 2
						4	Registers specified by regs = 3
						5	Registers specified by regs = 4
						8	Registers specified by regs = 7
						9	Registers specified by regs = 8
						10	Registers specified by regs = 9
						11	Registers specified by regs = 10
						12	Registers specified by regs = 11
	MOVM	regs	(SP)	S1	2	1	Registers specified by regs = 0
						1	Registers specified by regs = 1
						2	Registers specified by regs = 2
						3	Registers specified by regs = 3

Instruction		Source	Destination	Format	Code length	Execution Cycle	Remarks
MOVM						4	Registers specified by regs = 4
						8	Registers specified by regs = 7
						9	Registers specified by regs = 8
						10	Registers specified by regs = 9
						11	Registers specified by regs = 10
						12	Registers specified by regs = 11
CLR	CLR		Dn	S0	1	1	
ADD	ADD	Dm	Dn	S0	1	1	
	ADD	Dm	An	D0	2	1	
	ADD	Am	Dn	D0	2	1	
	ADD	Am	An	D0	2	1	
	ADD	imm8	Dn	S1	2	1	
	ADD	imm16	Dn	D2	4	1	
	ADD	imm32	Dn	D4	6	2	
	ADD	imm8	An	S1	2	1	
	ADD	imm16	An	D2	4	1	
	ADD	imm32	An	D4	6	2	
	ADD	imm8	SP	D1	3	1	
	ADD	imm16	SP	D2	4	1	
	ADD	imm32	SP	D4	6	2	
ADDC	ADDC	Dm	Dn	D0	2	1	
SUB	SUB	Dm	Dn	D0	2	1	
	SUB	Dm	An	D0	2	1	
	SUB	Am	Dn	D0	2	1	
	SUB	Am	An	D0	2	1	
	SUB	imm32	Dn	D4	6	2	
	SUB	imm32	An	D4	6	2	
SUBC	SUBC	Dm	Dn	D0	2	1	
MUL	MUL	Dm	Dn	D0	2	3	Dn=0
						13	Dn is a value which can be expressed with 1 byte
						21	Dn is a value which can be expressed with 2 bytes
						29	Dn is a value which can be expressed with 3 bytes
						34	Dn is a value which can be expressed with 4 bytes
MULU	MULU	Dm	Dn	D0	2	3	Dn=0
						13	Dn is a value which can be expressed with 1 byte
						21	Dn is a value which can be expressed with 2 bytes
						29	Dn is a value which can be expressed with 3 bytes
						34	Dn is a value which can be expressed with 4 bytes
DIV	DIV	Dm	Dn	D0	2	4	{MDR,Dn}=0
						14	{MDR, Dn} is a value which can be expressed with 1 byte
						22	{MDR, Dn} is a value which can be expressed with 2 bytes
						30	{MDR, Dn} is a value which can be expressed with 3 bytes
						38	{MDR, Dn} is a value which can be expressed with 4 bytes or more
DIVU	DIVU	Dm	Dn	D0	2	4	{MDR,Dn}=0
						14	{MDR, Dn} is a value which can be expressed with 1 byte
						22	{MDR, Dn} is a value which can be expressed with 2 bytes
						30	{MDR, Dn} is a value which can be expressed with 3 bytes
						38	{MDR, Dn} is a value which can be expressed with 4 bytes or more
INC	INC		Dn	S0	1	1	
	INC		An	S0	1	1	
INC4	INC4		An	S0	1	1	
CMP	CMP	Dm	Dn	S0	1	1	
	CMP	Dm	An	D0	2	1	
	CMP	Am	Dn	D0	2	1	
	CMP	Am	An	S0	1	1	
	CMP	imm8	Dn	S1	2	1	
	CMP	imm16	Dn	D2	4	1	
	CMP	imm32	Dn	D4	6	2	
	CMP	imm8	An	S1	2	1	
	CMP	imm16	An	D2	4	1	
	CMP	imm32	An	D4	6	2	

Instruction		Source	Destination	Format	Code length	Execution Cycle	Remarks
AND	AND	Dm	Dn	D0	2	1	
	AND	imm8	Dn	D1	3	1	
	AND	imm16	Dn	D2	4	1	
	AND	imm32	Dn	D4	6	2	
	AND	imm16	PSW	D2	4	1	
OR	OR	Dm	Dn	D0	2	1	
	OR	imm8	Dn	D1	3	1	
	OR	imm16	Dn	D2	4	1	
	OR	imm32	Dn	D4	6	2	
	OR	imm16	PSW	D2	4	1	
XOR	XOR	Dm	Dn	D0	2	1	
	XOR	imm16	Dn	D2	4	1	
	XOR	imm32	Dn	D4	6	2	
NOT	NOT		Dn	D0	2	1	
BTST	BTST	imm8	Dn	D1	3	1	
	BTST	imm16	Dn	D2	4	1	
	BTST	imm32	Dn	D4	6	2	
	BTST	imm8	(d8,An)	D2	4	4	
	BTST	imm8	(abs32)	D5	7	5	
BSET	BSET	Dm	(An)	D0	2	5	
	BSET	imm8	(d8,An)	D2	4	5	
	BSET	imm8	(abs32)	D5	7	6	
BCLR	BCLR	Dm	(An)	D0	2	5	
	BCLR	imm8	(d8,An)	D2	4	5	
	BCLR	imm8	(abs32)	D5	7	6	
ASR	ASR	Dm	Dn	D0	2	1	
	ASR	imm8	Dn	D1	3	1	
LSR	LSR	Dm	Dn	D0	2	1	
	LSR	imm8	Dn	D1	3	1	
ASL	ASL	Dm	Dn	D0	2	1	
	ASL	imm8	Dn	D1	3	1	
ASL2	ASL2		Dn	S0	1	1	
ROR	ROR		Dn	D0	2	1	
ROL	ROL		Dn	D0	2	1	
Bcc	BEQ	(d8,PC)		S1	2	3/1*	Branching taken/not taken
	BNE	(d8,PC)		S1	2	3/1*	Branching taken/not taken
	BGT	(d8,PC)		S1	2	3/1*	Branching taken/not taken
	BGE	(d8,PC)		S1	2	3/1*	Branching taken/not taken
	BLE	(d8,PC)		S1	2	3/1*	Branching taken/not taken
	BLT	(d8,PC)		S1	2	3/1*	Branching taken/not taken
	BHI	(d8,PC)		S1	2	3/1*	Branching taken/not taken
	BCC	(d8,PC)		S1	2	3/1*	Branching taken/not taken
	BLS	(d8,PC)		S1	2	3/1*	Branching taken/not taken
	BCS	(d8,PC)		S1	2	3/1*	Branching taken/not taken
	BVC	(d8,PC)		D1	3	4/2*	Branching taken/not taken
	BVS	(d8,PC)		D1	3	4/2*	Branching taken/not taken
	BNC	(d8,PC)		D1	3	4/2*	Branching taken/not taken
	BNS	(d8,PC)		D1	3	4/2*	Branching taken/not taken
	BRA	(d8,PC)		S1	2	3	Branching taken
Lcc	LEQ			S0	1	1/2*	Branching taken/not taken
	LNE			S0	1	1/2*	Branching taken/not taken
	LGT			S0	1	1/2*	Branching taken/not taken
	LGE			S0	1	1/2*	Branching taken/not taken
	LLE			S0	1	1/2*	Branching taken/not taken
	LLT			S0	1	1/2*	Branching taken/not taken
	LHI			S0	1	1/2*	Branching taken/not taken
	LCC			S0	1	1/2*	Branching taken/not taken
	LLS			S0	1	1/2*	Branching taken/not taken
	LCS			S0	1	1/2*	Branching taken/not taken
	LRA			S0	1	1	Branching taken

* Varies according to the state of the instruction buffer.

Instruction		Source	Destination	Format	Code length	Execution Cycle	Remarks
SETLB	SETLB			S0	1	1	
JMP	JMP	(An)		D0	2	3	
	JMP	(d16,PC)		S2	3	2	
	JMP	(d32,PC)		S4	5	4	
CALL	CALL	(d16,PC)	regs,imm8	S4	5	2	Registers specified by regs = 0
						3	Registers specified by regs = 1
						4	Registers specified by regs = 2
						5	Registers specified by regs = 3
						6	Registers specified by regs = 4
						9	Registers specified by regs = 7
						10	Registers specified by regs = 8
						11	Registers specified by regs = 9
						12	Registers specified by regs = 10
						13	Registers specified by regs = 11
CALL	CALL	(d32,PC)	regs,imm8	S6	7	5	Registers specified by regs = 0
						5	Registers specified by regs = 1
						6	Registers specified by regs = 2
						7	Registers specified by regs = 3
						8	Registers specified by regs = 4
						11	Registers specified by regs = 7
						12	Registers specified by regs = 8
						13	Registers specified by regs = 9
						14	Registers specified by regs = 10
						15	Registers specified by regs = 11
CALLS	CALLS	(An)		D0	2	3	
	CALLS	(d16,PC)		D2	4	3	
	CALLS	(d32,PC)		D4	6	4	
RET	RET	regs,imm8		S2	3	4	Registers specified by regs = 0
						4	Registers specified by regs = 1
						4	Registers specified by regs = 2
						4	Registers specified by regs = 3
						5	Registers specified by regs = 4
						8	Registers specified by regs = 7
						9	Registers specified by regs = 8
						10	Registers specified by regs = 9
						11	Registers specified by regs = 10
						12	Registers specified by regs = 11
RETF	RETF	regs,imm8		S2	3	2	Registers specified by regs = 0
						2	Registers specified by regs = 1
						3	Registers specified by regs = 2
						4	Registers specified by regs = 3
						5	Registers specified by regs = 4
						8	Registers specified by regs = 7
						9	Registers specified by regs = 8
						10	Registers specified by regs = 9
						11	Registers specified by regs = 10
						12	Registers specified by regs = 11
RETS	RETS			D0	2	4	
RTI	RTI			D0	2	4	
TRAP	TRAP			D0	2	4	
NOP	NOP			S0	1	1	
UDF	UDF20~35	Dm	Dn	D0	2	User defined	
	UDF00~15	Dm	Dn	D0	2	User defined	
	UDF00~15	imm8	Dn	D1	3	User defined	
	UDF00~15	imm16	Dn	D2	4	User defined	
	UDF00~15	imm32	Dn	D4	6	User defined	
	UDFU00~15	imm8	Dn	D1	3	User defined	
	UDFU00~15	imm16	Dn	D2	4	User defined	
	UDFU00~15	imm32	Dn	D4	6	User defined	

List of Extension Instructions (Code Length, Execution Cycle)

Instruction		Source	Destination	Format	Code length	Execution cycle	Remarks
PUTX	PUTX	Dm		D0	2	2	
	PUTCX	Dm	Dn	D0	2	2	
GETX	GETX		Dn	D0	2	2	
	GETCHX		Dn	D0	2	2	
	GETCLX		Dn	D0	2	2	
CLRMAC	CLRMAC			D0	2	2	
MULQ	MULQ	Dm	Dn	D0	2	4	Dm is a value which can be expressed with 2 bytes to 1 byte or Dm = 0
						5	Dm is a value which can be expressed with 4 bytes to 3 bytes
	MULQI	imm8	Dn	D1	3	4	
	MULQI	imm16	Dn	D2	4	4	
	MULQI	imm32	Dn	D4	6	5	imm is a value which can be expressed with 2 bytes to 1 byte or imm = 0
						6	imm is a value which can be expressed with 4 bytes to 3 bytes
MULQU	MULQU	Dm	Dn	D0	2	4	Dm is a value which can be expressed with 2 bytes to 1 byte or Dm = 0
						5	Dm is a value which can be expressed with 4 bytes to 3 bytes
	MULQIU	imm8	Dn	D1	3	4	
	MULQIU	imm16	Dn	D2	4	4	
	MULQIU	imm32	Dn	D4	6	5	imm is a value which can be expressed with 2 bytes to 1 byte or imm = 0
						6	imm is a value which can be expressed with 4 bytes to 3 bytes
MAC	MAC	Dm	Dn	D0	2	2	
	MACH	Dm	Dn	D0	2	2	
	MACB	Dm	Dn	D0	2	2	
MACU	MACU	Dm	Dn	D0	2	2	
	MACHU	Dm	Dn	D0	2	2	
	MACBU	Dm	Dn	D0	2	2	
SAT16	SAT16	Dm	Dn	D0	2	2	
SAT24	SAT24	Dm	Dn	D0	2	2	
MCST	MCST	Dm	Dn	D0	2	2	
	MCST	imm8	Dn	D0	2	2	
	MCST9		Dn	D0	2	2	
	MCST48		Dn	D0	2	2	
BSCH	BSCH	Dm	Dn	D0	2	2	
SWAP	SWAP	Dm	Dn	D0	2	2	
	SWAPH	Dm	Dn	D0	2	2	

Appendix C. Memory Connection Example

Fig. C-1 shows a connection example for the memory configuration described below.

Block 0:	16-bit bus,	4-Mbit ROM (262 144 words x 16 bits)
Block 1:	16-bit bus,	4-Mbit DRAM (262 144 words x 16 bits, 2 $\overline{\text{CAS}}$ control)
Block 2:	8-bit bus,	1-Mbit SRAM (131 072 words x 8 bits)

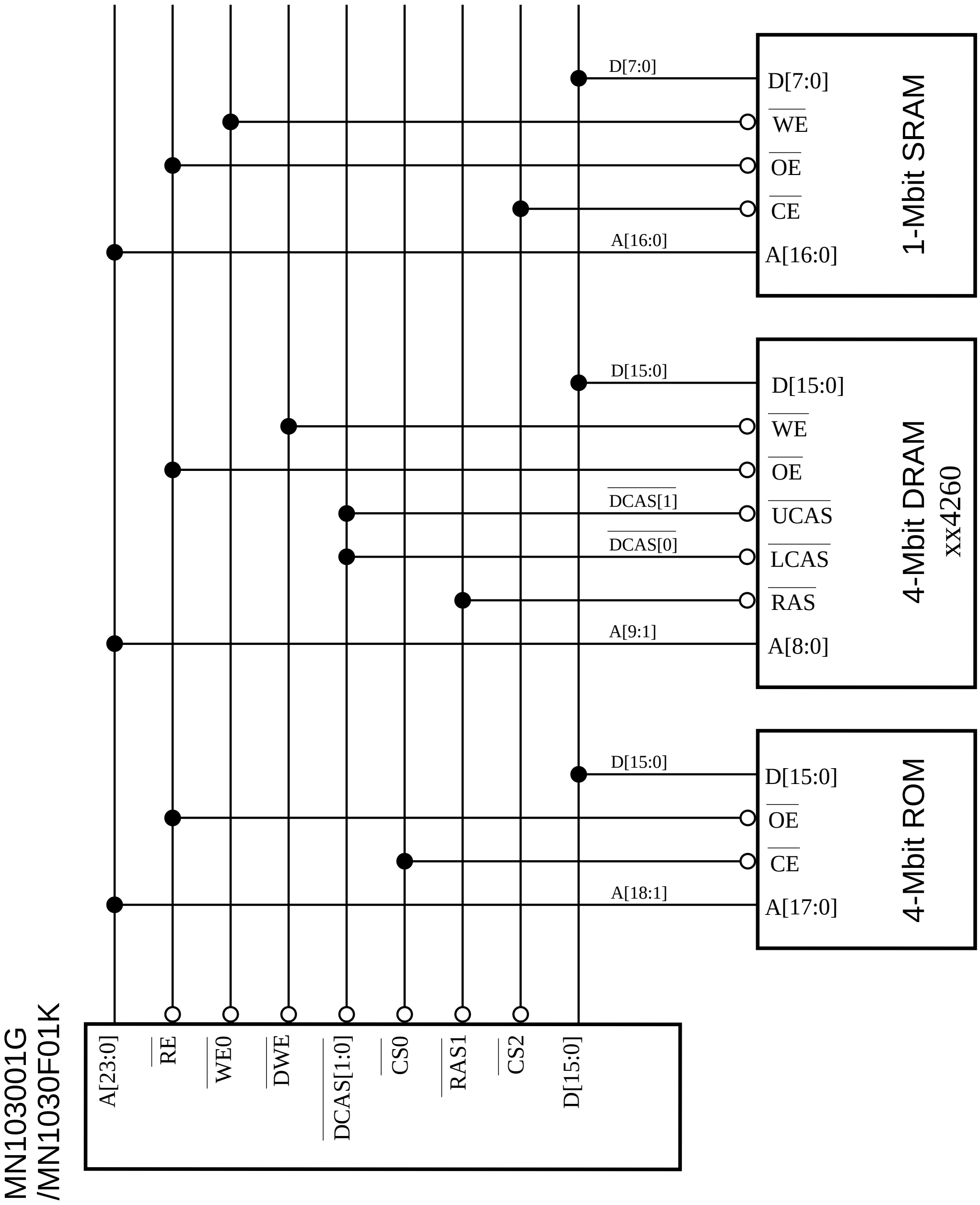


Fig. C-1 Memory Connection Example

Note : Fig. C-1 is provided as a reference example, and is not intended to guarantee operation.

Appendix D. Pins and Their Operating Statuses upon Reset

In the address/data separate mode

Pin No.	Pin name	Operating status	Pin No.	Pin name	Operating status	Pin No.	Pin name	Operating status	Pin No.	Pin name	Operating status
1	A19	L	26	PVSS	—	51	$\overline{\text{CS3}}$	H	76	P30	Hi-Z
2	A18	L	27	PVDD	—	52	$\overline{\text{CS2}}$	H	77	D15	Pull-Up
3	VDD	—	28	MMOD1	Input	53	$\overline{\text{CS1}}$	H	78	D14	Pull-Up
4	A17	L	29	MMOD0	Input	54	VDD	—	79	VDD	—
5	A16	L	30	$\overline{\text{RST}}$	Input	55	$\overline{\text{CS0}}$	H	80	D13	Pull-Up
6	A15	L	31	OSCO	Operating	56	P63	Hi-Z	81	D12	Pull-Up
7	A14	L	32	OSCI	Input	57	P62	Hi-Z	82	D11	Pull-Up
8	A13	L	33	VDD	—	58	P61	Hi-Z	83	D10	Pull-Up
9	VSS	—	34	SYSCLK	L	59	P60	Hi-Z	84	D9	Pull-Up
10	A12	L	35	VSS	—	60	VSS	—	85	VSS	—
11	A11	L	36	P96	Hi-Z	61	$\overline{\text{NMIRQ}}$	Hi-Z	86	VDD2 (VPP)*	—
12	A10	L	37	P95	Hi-Z	62	P55	Hi-Z	87	D8	Pull-Up
13	A9	L	38	$\overline{\text{WE1}}$	H	63	P54	Hi-Z	88	D7	Pull-Up
14	A8	L	39	$\overline{\text{WE0}}$	H	64	P53	Hi-Z	89	D6	Pull-Up
15	VDD	—	40	$\overline{\text{RE}}$	H	65	P52	Hi-Z	90	D5	Pull-Up
16	A7	L	41	VDD	—	66	P51	Hi-Z	91	D4	Pull-Up
17	A6	L	42	EXMOD1	Input	67	P50	Hi-Z	92	VDD	—
18	A5	L	43	EXMOD0	Input	68	P45	Hi-Z	93	D3	Pull-Up
19	A4	L	44	AVSS	—	69	VDD	—	94	D2	Pull-Up
20	A3	L	45	AN3	Hi-Z	70	P44	Hi-Z	95	D1	Pull-Up
21	VSS	—	46	AN2	Hi-Z	71	P43	Hi-Z	96	D0	Pull-Up
22	A2	L	47	AN1	Hi-Z	72	VSS	—	97	A22	L
23	A1	L	48	AN0	Hi-Z	73	P42	Hi-Z	98	VSS	—
24	A0	L	49	VREFH	—	74	P41	Hi-Z	99	A21	L
25	CKSEL	Input	50	AVDD	—	75	P40	Hi-Z	100	A20	L

Note 1) Hi-Z: High impedance
H: High level output
L: Low level output
Pull-up: Pull-up
Input: Input an appropriate value.

Note 2) The pin marked with an asterisk is VDD2 in the MN103001G, and VPP in the MN1030F01K.

In the address/data multiplex mode

Pin No.	Pin name	Operating status	Pin No.	Pin name	Operating status	Pin No.	Pin name	Operating status	Pin No.	Pin name	Operating status
1	A19	L	26	PVSS	—	51	$\overline{\text{CS3}}$	H	76	P30	Hi-Z
2	A18	L	27	PVDD	—	52	$\overline{\text{CS2}}$	H	77	P27	Hi-Z
3	VDD	—	28	MMOD1	Input	53	$\overline{\text{CS1}}$	H	78	P26	Hi-Z
4	A17	L	29	MMOD0	Input	54	VDD	—	79	VDD	—
5	A16	L	30	$\overline{\text{RST}}$	Input	55	$\overline{\text{CS0}}$	H	80	P25	Hi-Z
6	ADM15	Pull-Up	31	OSCO	Operating	56	P63	Hi-Z	81	P24	Hi-Z
7	ADM14	Pull-Up	32	OSCI	Input	57	P62	Hi-Z	82	P23	Hi-Z
8	ADM13	Pull-Up	33	VDD	—	58	P61	Hi-Z	83	P22	Hi-Z
9	VSS	—	34	SYSCLK	L	59	P60	Hi-Z	84	P21	Hi-Z
10	ADM12	Pull-Up	35	VSS	—	60	VSS	—	85	VSS	—
11	ADM11	Pull-Up	36	P96	Hi-Z	61	$\overline{\text{NMIRQ}}$	Hi-Z	86	VDD2(VPP)*	—
12	ADM10	Pull-Up	37	P95	Hi-Z	62	P55	Hi-Z	87	P20	Hi-Z
13	ADM9	Pull-Up	38	$\overline{\text{WE1}}$	H	63	P54	Hi-Z	88	P17	Hi-Z
14	ADM8	Pull-Up	39	$\overline{\text{WE0}}$	H	64	P53	Hi-Z	89	P16	Hi-Z
15	VDD	—	40	$\overline{\text{RE}}$	H	65	P52	Hi-Z	90	P15	Hi-Z
16	ADM7	Pull-Up	41	VDD	—	66	P51	Hi-Z	91	P14	Hi-Z
17	ADM6	Pull-Up	42	EXMOD1	Input	67	P50	Hi-Z	92	VDD	—
18	ADM5	Pull-Up	43	EXMOD0	Input	68	P45	Hi-Z	93	P13	Hi-Z
19	ADM4	Pull-Up	44	AVSS	—	69	VDD	—	94	P12	Hi-Z
20	ADM3	Pull-Up	45	AN3	Hi-Z	70	P44	Hi-Z	95	RWSEL	L
21	VSS	—	46	AN2	Hi-Z	71	P43	Hi-Z	96	AS	L
22	ADM2	Pull-Up	47	AN1	Hi-Z	72	VSS	—	97	A22	L
23	ADM1	Pull-Up	48	AN0	Hi-Z	73	P42	Hi-Z	98	VSS	—
24	ADM0	Pull-Up	49	VREFH	—	74	P41	Hi-Z	99	A21	L
25	CKSEL	Input	50	AVDD	—	75	P40	Hi-Z	100	A20	L

Note 1) Hi-Z: High impedance
H: High level output
L: Low level output
Pull-up: Pull-up
Input: Input an appropriate value.

Note 2) The pin marked with an asterisk is VDD2 in the MN103001G, and VPP in the MN1030F01K.

Appendix E. Package Outline

The package outline and dimensions of this microcontroller are shown below.

Package code : LQFP100-P-1414

Unit: mm

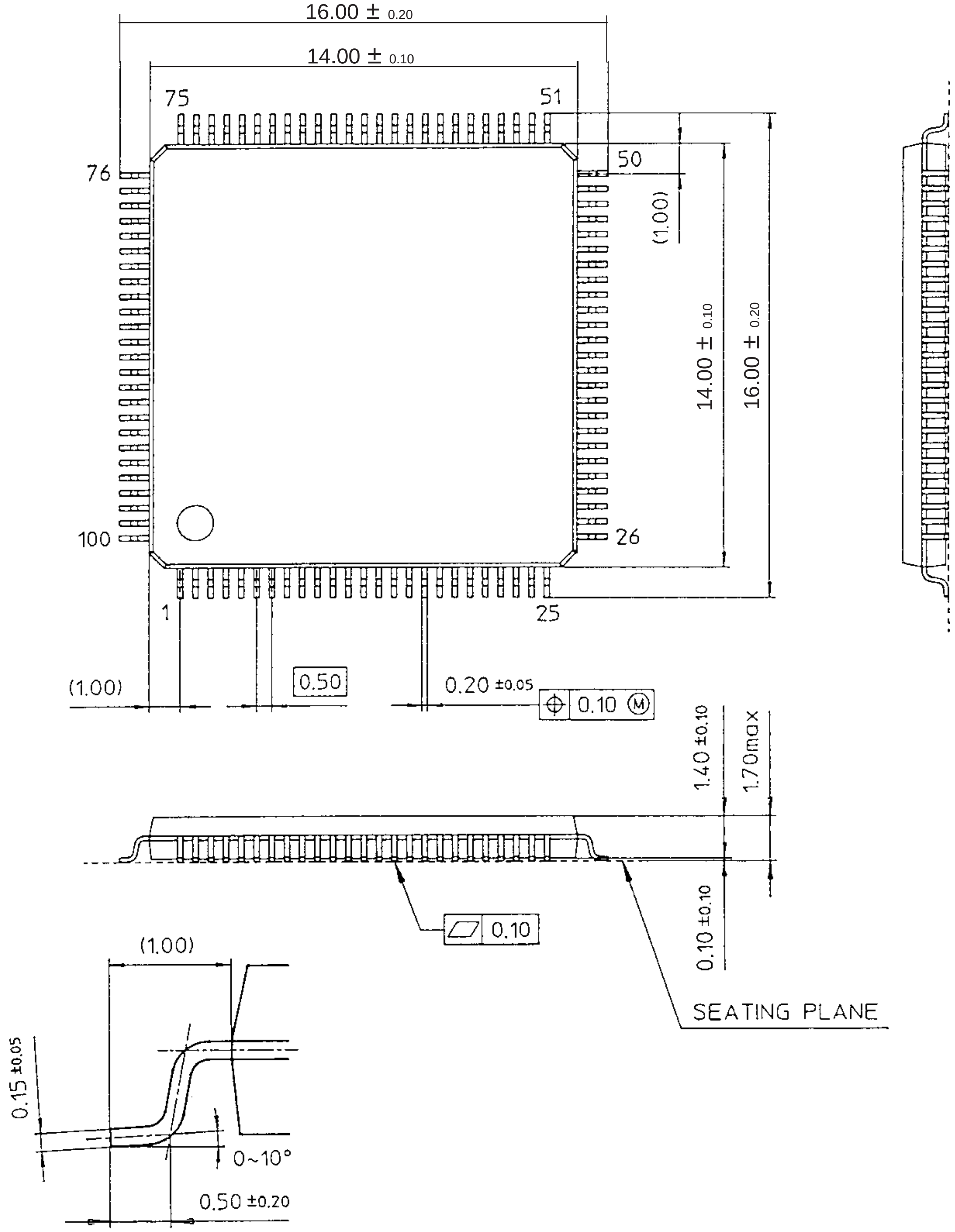


Fig. E-1 Package Outline

The correction table in The Revised Edition of MN103001G/F01K LSI User's Manual
(From 2nd Edition (or 2nd Edition 1st printing) to 5th Edition)

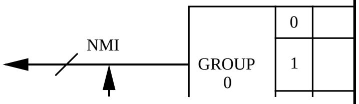
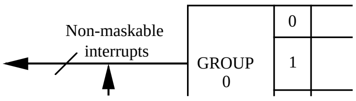
Page	Errors	Page	Corrections
P.1-3	- External interrupts: 9 sources (8 individual IRQs, and 1 external NMI)	P.1-3	- External interrupts: 9 sources (IRQn (n=7 to 0) x 8, and $\overline{\text{NMIRQ}}$ x 1)
P.1-8	(The column of "Pin Function" such as "Pin name" is "$\overline{\text{NMIRQ}}$" in the table.) External NMI input	P.1-8	(The column of "Pin Function" such as "Pin name" is "$\overline{\text{NMIRQ}}$" in the table.) External non-maskable interrupt input
P.2-9		P.2-9	(Following sentence is added.) If the CPUM register is accessed to make a transition to an operating mode of SLEEP/HALT/STOP during execution of a program in external memory, a branch instruction should not be located within the three instructions immediately following the CPUM register access instruction.
P.2-13	(Note) Interrupts are prohibited and the bus is locked (occupied by the CPU) when executing BSET or BCLR.	P.2-13	Note : Interrupts are prohibited and the bus is locked (occupied by the CPU) when executing BSET or BCLR, however, if a BSET or BCLR instruction is executed during program execution in external memory, a bus authority release due to an external bus request may be interposed between the data read and data write by the BSET or BCLR instruction. If the atomic bus cycles (i.e. bus lock) of the BSET or BCLR instruction need to be guaranteed in a system that uses multiple processors, either of the following measures should be taken. 1. A program in which a BSET or BCLR instruction is executed should be placed in internal memory. 2. Designate the bus authority request pin ($\overline{\text{BR}}$) as a general-purpose input port, and the bus authority release pin ($\overline{\text{BG}}$) as a general-purpose output port, for instance, so that bus requests cannot be accepted during execution of a BSET or BCLR instruction.
P.2-14	(In the figure) Reset interrupt NMI interrupt	P.2-14	(In the figure) Reset interrupt Non-maskable interrupt
P.2-15	(In the table 2-5-1) Interrupt prohibited (only NMI accepted)	P.2-15	(In the table 2-5-1) Interrupt prohibited (only non-maskable interrupts accepted)
P.2-15	(In the 4th line of [Interrupt Control Registers(GnICR)]) Register G0ICR is dedicated for NMI interrupts, and ...	P.2-15	(In the 4th line of [Interrupt Control Registers(GnICR)]) Register G0ICR is dedicated for non-maskable interrupts, and ...
P.2-17	(In the 9th line of [Interrupt Accept Group Register(IAGR)]) Accessing IAGR is meaningless during NMI interrupts.	P.2-17	(In the 9th line of [Interrupt Accept Group Register(IAGR)]) Accessing IAGR is meaningless during non-maskable interrupts.
P.2-18	(In the 2nd line of "External pin non-maskable interrupt") ..., the external NMI request flag (NMIF) in the...	P.2-18	(In the 2nd line of "External pin non-maskable interrupt") ..., the external non-maskable interrupt request flag (NMIF) in the...
P.2-18	(The 1st line of "System error interrupt") System error interrupt occurs when an unimplemented instruction is executed or other fatal error occurs.	P.2-18	(The 1st line of "System error interrupt") System error interrupt occurs when an unaligned memory access or an unimplemented instruction is executed or other fatal error occurs.
P.2-18		P.2-18	(Following note is added.) Note: Do not change the interrupt enable (IE) in PSW during non-maskable interrupt processing.
P.2-19	(*Interrupt processing sequences executed by the hardware*) 1. The PC (return address) is saved to the stack (SP-4). 2. The PSW is saved to the stack (SP-8).	P.2-19	(*Interrupt processing sequences executed by the hardware*) 1. The PSW is saved to the stack (SP-8). 2. The PC (return address) is saved to the stack (SP-4).
P.2-19	(In the 4th line of "*Interrupt processing sequences executed by the hardware*") (IM2 to IM0 is undefined in case of NMI.)	P.2-19	(In the 4th line of "*Interrupt processing sequences executed by the hardware*") (IM2 to IM0 is undefined in case of non-maskable interrupts.)
P.2-20	(In the number 3 of "*Example of pre-processing by the interrupt handler*") * In case of NMI, ...	P.2-20	(In the number 3 of "*Example of pre-processing by the interrupt handler*") * In case of non-maskable interrupts, ...
P.3-5		P.3-5	([Programming Cautions] is added for PUTX.)
P.3-7		P.3-7	(Following sentences are added to [Programming Cautions] of GETX.) When "udf15 Dm, Dn" is operated, Dm is ignored. The operations of "udf15 imm8, Dn", "udf15 imm16, Dn" and "udf15 imm32, Dn" are not assured. In addition, a system error interrupt does not occur in these cases.

Page	Errors	Page	Corrections
P.3-8		P.3-8	(Following sentences are added to [Programming Cautions] of GETCHX.) When "udf12 Dm, Dn" is operated, Dm is ignored. The operations of "udf12 imm8, Dn", "udf12 imm16, Dn" and "udf12 imm32, Dn" are not assured. In addition, a system error interrupt does not occur in these cases.
P.3-9		P.3-9	(Following sentences are added to [Programming Cautions] of GETCLX.) When "udf13 Dm, Dn" is operated, Dm is ignored. The operations of "udf13 imm8, Dn", "udf13 imm16, Dn" and "udf13 imm32, Dn" are not assured. In addition, a system error interrupt does not occur in these cases.
P.3-10		P.3-10	([Programming Cautions] is added for CLRMAC.)
P.3-21		P.3-21	(Following sentence is added to [Programming Cautions] of SAT16.) The operations of "udf04 imm8, Dn", "udf04 imm16, Dn" and "udf04 imm32, Dn" are not assured. In addition, a system error interrupt does not occur in these cases.
P.3-22		P.3-22	(Following sentence is added to [Programming Cautions] of SAT24.) The operations of "udf05 imm8, Dn", "udf05 imm16, Dn" and "udf05 imm32, Dn" are not assured. In addition, a system error interrupt does not occur in these cases.
P.3-23	[Instruction Format (Macro Name)] MCST32, MCST16, MCST8	P.3-23	[Instruction Format (Macro Name)] MCST Dm, Dn MCST imm8, Dn
P.3-23	(From 2nd line of [Operation]) In addition, depending on the value of Dm, ... (1) When the value of Dm is 32 (0x00000020) : (2) When the value of Dm is 16 (0x00000010) : (3) When the value of Dm is 8 (0x00000008) : (4) When the value of Dm is any other value	P.3-23	(From 2nd line of [Operation]) In addition, depending on the value of Dm or imm8, ... (1) When the value of Dm or imm8 is 32 (0x00000020) : (2) When the value of Dm or imm8 is 16 (0x00000010) : (3) When the value of Dm or imm8 is 8 (0x00000008) : (4) When the value of Dm or imm8 is any other value
P.3-24		P.3-24	(Following sentence is added to [Programming Cautions] of MCST.) The operations of "udf02 imm16, Dn" and "udf02 imm32, Dn" are not assured. In addition, a system error interrupt does not occur in these cases.
P.3-25	[Instruction Format (Macro Name)] MCST9 Dn, Dn	P.3-25	[Instruction Format (Macro Name)] MCST9 Dn
P.3-25		P.3-25	(Following sentences are added to [Programming Cautions] of MCST9.) When "udf03 Dm, Dn" is operated, Dm is ignored. The operations of "udf03 imm8, Dn", "udf03 imm16, Dn" and "udf03 imm32, Dn" are not assured. In addition, a system error interrupt does not occur in these cases.
P.3-26	[Instruction Format (Macro Name)] MCST48 Dn, Dn	P.3-26	[Instruction Format (Macro Name)] MCST48 Dn
P.3-26		P.3-26	(Following sentences are added to [Programming Cautions] of MCST48.) When "udf06 Dm, Dn" is operated, Dm is ignored. The operations of "udf06 imm8, Dn", "udf06 imm16, Dn" and "udf06 imm32, Dn" are not assured. In addition, a system error interrupt does not occur in these cases.
P.3-27		P.3-27	(Following sentence is added to [Programming Cautions] of BSCH.) The operations of "udf07 imm8, Dn", "udf07 imm16, Dn" and "udf07 imm32, Dn" are not assured. In addition, a system error interrupt does not occur in these cases.
P.3-29		P.3-29	(Following sentence is added to [Programming Cautions] of SWAP.) The operations of "udf08 imm8, Dn", "udf08 imm16, Dn" and "udf08 imm32, Dn" are not assured. In addition, a system error interrupt does not occur in these cases.

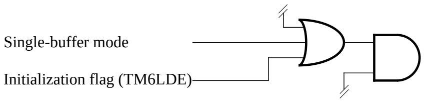
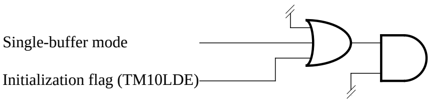
<table>
<tr><th>Page</th><th>Errors</th><th>Page</th><th>Corrections</th></tr>
<tr><td>P.3-30</td><td></td><td>P.3-30</td><td>(Following sentence is added to [Programming Cautions] of SWAPH.)
The operations of "udf09 imm8, Dn", "udf09 imm16, Dn" and "udf09 imm32, Dn" are not assured. In addition, a system error interrupt does not occur in these cases.</td></tr>
<tr><td>P.3-31</td><td>(Omit)</td><td>P.3-31</td><td>(One note is added in the table 3-2-1.)
<table>
<tr><th>Preceding instruction</th><th>Following instruction</th><th>Placement relationship</th><th>Notes</th></tr>
<tr><td>:</td><td>:</td><td>:</td><td>:</td></tr>
<tr><td>(◆ For details, refer to note (e) on page 3-36.)</td><td>Multiply-and-accumulate instruction *3
High -speed multiplication instruction *5</td><td>-</td><td>Insert at least two NOP instructions immediately before the instructions</td></tr>
</table></td></tr>
<tr><td>–</td><td></td><td>P.3-36 to P.3-39</td><td>(As the programming note, a new item (e) is added. For details on this item, refer to this LSI User's Manual.)</td></tr>
<tr><td>P.4-4</td><td>Note that operation is not assured when attempting to access unmounted space.</td><td>P.4-4</td><td>Note that it is prohibited to access unmounted space of the internal data space, the internal I/O space and the internal instruction space. When accessing the unmounted space, the operation is not assured.</td></tr>
<tr><td>P.4-5</td><td>Note that operation is not assured when attempting to access unmounted space.</td><td>P.4-5</td><td>Note that it is prohibited to access unmounted space of the internal data space and the internal I/O space. When accessing the unmounted space, the operation is not assured.</td></tr>
<tr><td>P.5-4</td><td>(The 2nd line of "■ Operation of various peripheral functions in the low power consumption modes".)
In SLEEP mode, all peripheral functions operate except for the bus controller.</td><td>P.5-4</td><td>(The 2nd line of "■ Operation of various peripheral functions in the low power consumption modes".)
In SLEEP mode, all peripheral functions operate except for the bus controller and the watchdog timer.</td></tr>
<tr><td>P.5-4</td><td>(The 5th line of "■ Operation of various peripheral functions in the low power consumption modes".)
In HALT mode, the interrupt controller accepts interrupt requests from the watchdog timer and external pin interrupt requests, notifies the CPU core, and then initiates recovery from HALT mode. In STOP mode, the interrupt controller accepts external pin interrupt requests, notifies the CPU core, and then initiates recovery from STOP mode.</td><td>P.5-4</td><td>(The 5th line of "■ Operation of various peripheral functions in the low power consumption modes".)
In HALT/STOP mode, the interrupt controller accepts the external pin interrupt requests, notifies the CPU core, and then initiates recovery from HALT/STOP mode.
When making a transition to HALT/SLEEP mode, stop the watchdog timer by clearing the WDCNE bit to "0" in watchdog timer control register WDCTR.</td></tr>
<tr><td>P.5-4</td><td>
<table>
<tr><th></th><th>SLEEP</th><th>HALT</th><th>STOP</th></tr>
<tr><td>⋮</td><td>⋮</td><td>⋮</td><td>⋮</td></tr>
<tr><td>Watchdog timer</td><td>Operates</td><td>Operates</td><td>Stopped</td></tr>
<tr><td>⋮</td><td>⋮</td><td>⋮</td><td>⋮</td></tr>
</table></td><td>P.5-4</td><td>
<table>
<tr><th></th><th>SLEEP</th><th>HALT</th><th>STOP</th></tr>
<tr><td>⋮</td><td>⋮</td><td>⋮</td><td>⋮</td></tr>
<tr><td>Watchdog timer</td><td>Stopped</td><td>Stopped</td><td>Stopped</td></tr>
<tr><td>⋮</td><td>⋮</td><td>⋮</td><td>⋮</td></tr>
</table></td></tr>
<tr><td>P.6-2</td><td>(In "6.1 Overview")
... , the CG also supplies clock pulses with the same frequency as the oscillating frequency of the oscillator to external devices.</td><td>P.6-2</td><td>(In "6.1 Overview")
..., the CG also supplies clock pulses with the same frequency as the oscillating frequency of the oscillator, or that frequency divided by 2, to external devices.</td></tr>
<tr><td>P.6-2</td><td>(In the description of "■ Flexible clock control".)
- Supports self-excitation/external excitation
(input frequency: 8 to 30 MHz)</td><td>P.6-2</td><td>(In the description of "■ Flexible clock control".)
- Supports self-excitation/external excitation
(input frequency: 8 MHz to 20 MHz)
Note: The in-circuit emulator (ICE) cannot operate with self-excited oscillators in the microcontroller.</td></tr>
<tr><td>P.6-2</td><td>(In fig. 6-3-1.)
When using PLL:
8MHz to 15MHz
When not using PLL:
8MHz to 30MHz</td><td>P.6-2</td><td>(In fig. 6-3-1.)
When using PLL:
8 MHz to 18 MHz
When not using PLL:
8 MHz to 20 MHz</td></tr>
<tr><td>P.6-3</td><td>(From 1st line to 4th line of "6.4.1 Input Frequency Setting".)
When CKSEL is set “H”, use an oscillator or resonator with an input frequency fosci such that 8MHz ≦ fosci ≦ 15MHz. When CKSEL is set “L”, use an oscillator or resonator with an input frequency fosci such that 8MHz ≦ fosci ≦ 30MHz.
Use of an oscillator or resonator that generates a frequency lower than 8MHz, or higher than 30 MHz is prohibited.</td><td>P.6-3</td><td>(From 1st line to 4th line of "6.4.1 Input Frequency Setting".)
When CKSEL is set “H”, use an oscillator or resonator with an input frequency fosci such that 8 MHz ≦ fosci ≦ 18 MHz. When CKSEL is set “L”, use an oscillator or resonator with an input frequency fosci such that 8 MHz ≦ fosci ≦ 20 MHz.
Use of an oscillator or resonator that generates a frequency lower than 8 MHz, or higher than 20 MHz is prohibited.</td></tr>
</table>

<table>
<tr><th>Page</th><th>Errors</th><th>Page</th><th>Corrections</th></tr>
<tr><td>P.6-3</td><td><table><tr><th>Input frequency range</th><th>PLL</th></tr><tr><td>8 ≦ fosci ≦ 15 MHz</td><td>When using</td></tr><tr><td>8 ≦ fosci ≦ 30 MHz</td><td>When not using</td></tr></table></td><td>P.6-3</td><td><table><tr><th>Input frequency range</th><th>PLL</th></tr><tr><td>8 MHz ≦ fosci ≦ 18 MHz</td><td>When using</td></tr><tr><td>8 MHz ≦ fosci ≦ 20 MHz</td><td>When not using</td></tr></table></td></tr>
<tr><td>P.6-3</td><td>When the reset state is released, SYSCLK, MCLK, and IOCLK are supplied starting after a certain oscillation stabilization wait time.
Note: • When a clock is supplied from external, input the clock to the OSCI pin, and leave the OSCO pin open.
• For details on the oscillation stabilization wait time, refer to Chapter 12, “Watchdog Timer.”</td><td>P.6-3</td><td>When the reset state is released, SYSCLK, MCLK, and IOCLK are supplied starting after a certain oscillation stabilization wait time.
Note: For details on the oscillation stabilization wait time, refer to Chapter 12, “Watchdog Timer.”
Note 1: When a clock is supplied from external, input the clock to the OSCI pin, and leave the OSCO pin open.
Note 2: The in-circuit emulator (ICE) cannot operate with self-excited oscillators in the microcontroller. Use the clock generated in the target system.
When the clock is generated in the target system, supply the clock to the in-circuit emulator main unit through a buffer with adequate drive capability. The in-circuit emulator will not operate correctly if the amplitude of the clock is inadequate, the clock signal is noisy, or the buffer has inadequate drive capability.</td></tr>
<tr><td>P.6-4</td><td><table><tr><th>fosci(MHz)</th><th></th><th>Multiple</th><th>Frequency fsys(MHz)</th><th>Multiple</th><th>Frequency fc (MHz)</th><th>Multiple</th><th>Frequency fio (MHz)</th></tr><tr><td rowspan="3">8≦fosci≦15</td><td rowspan="3"></td><td rowspan="3">1</td><td rowspan="3">8≦fsys≦15</td><td>4</td><td>32≦fc≦60*</td><td>1</td><td>8≦fio≦15</td></tr><tr><td>2</td><td>16≦fc≦30</td><td>1/2</td><td>4≦fio≦7.5</td></tr><tr><td>1</td><td>8≦fc≦15</td><td>1/4</td><td>2≦fio≦3.75</td></tr><tr><td>8≦fosci≦30</td><td></td><td>1/2</td><td>4≦fsys≦15</td><td>1/2</td><td>4≦fc≦15</td><td>1/8</td><td>1≦fio≦3.75</td></tr></table></td><td>P.6-4</td><td><table><tr><th>fosci(MHz)</th><th></th><th>Multiple</th><th>Frequency fsys(MHz)</th><th>Multiple</th><th>Frequency fc (MHz)</th><th>Multiple</th><th>Frequency fio (MHz)</th></tr><tr><td>8≦fosci≦15</td><td></td><td>1</td><td>8≦fsys≦15</td><td>4</td><td>32≦fc≦60*</td><td>1</td><td>8≦fio≦15</td></tr><tr><td rowspan="2">8≦fosci≦18</td><td rowspan="2"></td><td rowspan="2">1</td><td rowspan="2">8≦fsys≦18</td><td>2</td><td>16≦fc≦36</td><td>1/2</td><td>4≦fio≦9</td></tr><tr><td>1</td><td>8≦fc≦18</td><td>1/4</td><td>2≦fio≦4.5</td></tr><tr><td>8≦fosci≦20</td><td></td><td>1/2</td><td>4≦fsys≦10</td><td>1/2</td><td>4≦fc≦10</td><td>1/8</td><td>1≦fio≦2.5</td></tr></table></td></tr>
<tr><td>P.8-5</td><td><table><tr><th>Function</th></tr><tr><td>Oscillator input pin (when using PLL: 8MHz to 15MHz; when not using PLL: 8MHz to 30MHz)</td></tr><tr><td>Oscillator output pin (when using PLL: 8MHz to 15MHz; when not using PLL: 8MHz to 30MHz)</td></tr></table></td><td>P.8-5</td><td><table><tr><th>Function</th></tr><tr><td>Oscillator input pin (when using PLL: 8 MHz to 18 MHz; when not using PLL: 8 MHz to 20 MHz)</td></tr><tr><td>Oscillator output pin (when using PLL: 8 MHz to 18 MHz; when not using PLL: 8 MHz to 20 MHz)</td></tr></table></td></tr>
<tr><td>P.8-11,
P.8-15</td><td></td><td>P.8-11,
P.8-15</td><td>(Following note is added under the table of "When Using DRAM")
Note: When performing ICE trace/emulation in software page mode, set the CAS parameter to a value of “5” or higher.</td></tr>
<tr><td>P.8-15,
P.8-20</td><td>(The Setting condition of EA1 to 0 bit in the table of "When Using handshaking mode")
00: 0MCLK
≀ ≀
11: 3MCLK</td><td>P.8-15,
P.8-20</td><td>(The Setting conditions of EA1 to 0 bit in the table of "When Using handshaking mode")
00: prohibited
01: 1MCLK
10: 2MCLK
11: 3MCLK</td></tr>
<tr><td>P.8-35</td><td></td><td>P.8-35</td><td>(Following sentence is added to 13th line.)
The $\overline{DK}$ signal connected to the microcontroller should be input so as to be asserted from point EA+DW onward, and is negated before the next access.</td></tr>
<tr><td>P.8-35</td><td></td><td>P.8-35</td><td>(In figure 8-13-4, the $\overline{DK}$ signal asserted by the read access was changed so as to be negated before the write access.)</td></tr>
<tr><td>P.8-36</td><td></td><td>P.8-36</td><td>(In figure 8-13-5 and 8-13-6, the $\overline{DK}$ signal asserted by the read access was changed so as to be negated before the write access. Moreover, the signal name, $\overline{CSn}$ was changed to $\overline{CS2}$.)</td></tr>
<tr><td>P.8-41</td><td></td><td>P.8-41</td><td>(Following sentence is added to 20th line.)
The $\overline{DK}$ signal connected to the microcontroller should be input so as to be asserted from point EA+DW onward, and is negated before the next access.</td></tr>
</table>

Page	Errors	Page	Corrections
P.8-42		P.8-42	(In figure 8-13-13 (a) and (b), the $\overline{\mathrm{DK}}$ signal asserted by the low-order side access was changed so as to be negated before the high-order side access.)
P.8-43 to P.8-44		P.8-43 to P.8-44	(In figure 8-13-14 (a), (b) and figure 8-13-15 (a), (b), the $\overline{\mathrm{DK}}$ signal asserted by the low- order side access was changed so as to be negated before the high-order side access. Moreover, the signal name, $\overline{\mathrm{CSn}}$ was changed to $\overline{\mathrm{CS2}}$.)
P.8-48		P.8-48	(Following sentence is added to 17th line.) The $\overline{\mathrm{DK}}$ signal connected to the microcontroller should be input so as to be asserted from point EA+DW onward, and is negated before the next access.
P.8-49		P.8-49	(In figure 8-13-20, the $\overline{\mathrm{DK}}$ signal was changed so as to be asserted from point EA+DW onward. The $\overline{\mathrm{DK}}$ signal asserted by the read access was changed so as to be negated before the write access.)
P.8-49		P.8-49	(In figure 8-13-21, the $\overline{\mathrm{DK}}$ signal was changed so as to be asserted from point EA+DW onward. The $\overline{\mathrm{DK}}$ signal asserted by the read access was changed so as to be negated before the write access. Moreover, the signal name, $\overline{\mathrm{CSn}}$ was changed to $\overline{\mathrm{CS2}}$.)
P.8-50		P.8-50	(In figure 8-13-22, the $\overline{\mathrm{DK}}$ signal asserted by the read access was changed so as to be negated before the write access. Moreover, the signal name, $\overline{\mathrm{CSn}}$ was changed to $\overline{\mathrm{CS2}}$.)
P.8-56		P.8-56	(Following sentence is added to 23th line.) The $\overline{\mathrm{DK}}$ signal connected to the microcontroller should be input so as to be asserted from point EA+DW onward, and is negated before the next access.
P.8-57		P.8-57	(In figure 8-13-27 (a) and (b), the $\overline{\mathrm{DK}}$ signal asserted by the low-order side access was changed so as to be negated before the high-order side access. Moreover, the figure was changed to one in the case that parameter values were EA=1 and DW=1.)
P.8-58		P.8-58	(In figure 8-13-28 (a) and (b), the $\overline{\mathrm{DK}}$ signal was changed so as to be asserted from point EA+DW onward. The $\overline{\mathrm{DK}}$ signal asserted by the low- order side access was changed so as to be negated before the high-order side access. Moreover, the signal name, $\overline{\mathrm{CSn}}$ was changed to $\overline{\mathrm{CS2}}$.)
P.8-59		P.8-59	(In figure 8-13-29 (a) and (b), the $\overline{\mathrm{DK}}$ signal asserted by the low-order side access was changed so as to be negated before the high-order side access. Moreover, the signal name, $\overline{\mathrm{CSn}}$ was changed to $\overline{\mathrm{CS2}}$.)
-		P.8-73 to P.8-74	(Following two cautions are added.) 5. Interrupts are prohibited and the bus is locked (occupied by the CPU) when executing BSET or BCLR, however, if a BSET or BCLR instruction is executed during program execution in external memory, a bus authority release due to an external bus request may be interposed between the data read and data write by the BSET or BCLR instruction. If the atomic bus cycles (i.e. bus lock) of the BSET or BCLR instruction need to be guaranteed in a system that uses multiple processors, either of the following measures should be taken. 1. A program in which a BSET or BCLR instruction is executed should be placed in internal memory. 2. Designate the bus authority request pin ($\overline{\mathrm{BR}}$) as a general-purpose input port, and the bus authority release pin ($\overline{\mathrm{BG}}$) as a general-purpose output port, for instance, so that bus requests cannot be accepted during execution of a BSET or BCLR instruction.

<table>
<tr><th>Page</th><th>Errors</th><th>Page</th><th>Corrections</th></tr>
<tr><td>P.9-3</td><td>(In fig. 9-4-1.)
 </td><td>P.9-3</td><td>(In fig. 9-4-1.)
 </td></tr>
<tr><td>P.9-7</td><td>(Register's purpose)
This register determines whether an NMI interrupt has been generated.</td><td>P.9-7</td><td>(Register's purpose)
This register determines whether a non-maskable interrupt has been generated.</td></tr>
<tr><td>P.9-7</td><td>Bit name Description
NMIF External NMI request flag</td><td>P.9-7</td><td>Bit name Description
NMIF External non-maskable interrupt request flag</td></tr>
<tr><td>P.9-7</td><td>(From 1st line of main sentence)
Each flag is set if the corresponding NMI interrupt request is generated.

After an NMI interrupt is accepted, clear it via the software from within the NMI interrupt processing program.

When a flag is set to “1”, write a “1” to the flag to clear it.</td><td>P.9-7</td><td>(From 1st line of main sentence)
The method of clearing flag differs according to the interrupt request flags.
1. External non-maskable interrupt request flag (NMIF) and Watchdog timer overflow interrupt request flag (WDIF)
After a non-maskable interrupt is accepted, these flags can be cleared by writing to the non-maskable interrupt control register (NMICR).
When a flag is set to “1”, write a “1” to the flag to clear it.</td></tr>
<tr><td>P.9-7</td><td>Note: An NMI cannot be generated through software.</td><td>P.9-7</td><td>Note: A non-maskable interrupt cannot be generated through software.</td></tr>
<tr><td>P.9-7</td><td></td><td>P.9-7</td><td>(Following sentence is added to under the table shown the change of flag.)
2. System error interrupt request flag (SYSEF)
This flag cannot be cleared by writing to the non-maskable interrupt control register (NMICR).
This flag can be cleared by generating a reset interrupt by setting the $\overline{\text{RST}}$ pin to "L" level or by the self-reset, which is generated by writing to the reset control register (RSTCTR) of the watchdog timer.</td></tr>
<tr><td>P.9-8</td><td><table>
<tr><th colspan="2">Write data</th><th colspan="2">Result of write</th></tr>
<tr><th>IRn</th><th>IDn</th><th>IRn</th><th>IDn</th></tr>
<tr><td>0</td><td>0</td><td>No change</td><td>No change</td></tr>
<tr><td>1</td><td>0</td><td>No change</td><td>No change</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>IEn value</td></tr>
</table>Note) n= 0, 1, 2, 3</td><td>P.9-8</td><td><table>
<tr><th colspan="2">Write data</th><th>Result of write</th></tr>
<tr><th>IRn</th><th>IDn</th><th>IRn</th></tr>
<tr><td>0</td><td>0</td><td>No change</td></tr>
<tr><td>1</td><td>0</td><td>No change</td></tr>
<tr><td>0</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>1</td></tr>
</table>Note) n= 0, 1, 2, 3
The value of IDn after the write is the logical product of the value of IEn after the write and the value of IRn after the write.</td></tr>
<tr><td>P.9-30</td><td>(The 3rd line right after the itemizations.)
..., it is determined to be either a non-maskable interrupt (NMI) ...</td><td>P.9-30</td><td>(The 3rd line right after the itemizations.)
..., it is determined to be either a non-maskable interrupt ...</td></tr>
<tr><td>P.9-30</td><td>(The 2nd line from the bottom.)
..., the NMI interrupt request is simply sent to the CPU.</td><td>P.9-30</td><td>(The 2nd line from the bottom.)
..., the non-maskable interrupt request is simply sent to the CPU.</td></tr>
<tr><td>P.11-6</td><td> </td><td>P.11-6</td><td> </td></tr>
<tr><td>P.12-2</td><td>When the CKSEL pin input is "H" (oscillating frequency: 8 to 15 MHz):
16, 18, 20, 22, or 24 bits can be selected.
When the CKSEL pin input is "L" (oscillating frequency: 8 to 30 MHz):</td><td>P.12-2</td><td>When the CKSEL pin input is "H" (oscillating frequency: 8 MHz to 18 MHz):
16, 18, 20, 22, or 24 bits can be selected.
When the CKSEL pin input is "L" (oscillating frequency: 8 MHz to 20 MHz):</td></tr>
<tr><td>P.12-2</td><td>Overflow cycle: 4.369 ms to 1118.481 ms
(when the CKSEL pin input is "H" and the oscillating frequency is 15 MHz)
4.369 ms to 1118.481 ms
(when the CKSEL pin input is "L" and the oscillating frequency is 30 MHz)</td><td>P.12-2</td><td>Overflow cycle: 4.369 ms to 1118.481 ms
(when the CKSEL pin input is "H" and the oscillating frequency is 15 MHz)</td></tr>
<tr><td>P.12-2</td><td>(The 2nd line from the bottom.)
When recovering from STOP mode: 4.369 ms to 1118.481 ms</td><td>P.12-2</td><td>(The 2nd line from the bottom.)
When recovering from STOP mode: 4.369 ms to 1118.481 ms
<Recommended value is 14 ms or longer.&gt;</td></tr>
</table>

<table>
<tr><th>Page</th><th>Errors</th><th>Page</th><th>Corrections</th></tr>
<tr><td>P.12-5</td><td>(In the table of Example.)<table><tr><td>Overflow cycle
When CKSEL is "H" and oscillation frequency is 15 MHz
(or when CKSEL is "L" and oscillating frequency is 30 MHz)</td></tr></table></td><td>P.12-5</td><td>(In the table of Example.)<table><tr><td>Overflow cycle
When CKSEL is "H" and oscillation frequency is 15 MHz</td></tr></table></td></tr>
<tr><td>P.12-7</td><td>(The 2nd line from the bottom.)
An oscillation stabilization wait time of at least 17ms is recommended.</td><td>P.12-7</td><td>(The 2nd line from the bottom.)
An oscillation stabilization wait time of at least 14 ms is recommended.</td></tr>
<tr><td>P.12-8</td><td>(In fig.12-5-2.)
4.369 ms to 1118.481 ms</td><td>P.12-8</td><td>(In fig.12-5-2.)
4.369 ms to 1118.481 ms <Recommended value is 14 ms or longer.&gt;</td></tr>
<tr><td>P.13-43</td><td><table><tr><th>Bit No.</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr><tr><td>Bit name</td><td>SC3 TXF</td><td>SC3 RXF</td><td>SC3 TBF</td><td>SC3 RBF</td><td>SC3 CTS</td><td>SC3 FEF</td><td>SC3 PEF</td><td>SC3 OEF</td></tr><tr><td>Reset</td><td>0</td><td>0</td><td>0</td><td>0</td><td>*1</td><td>0</td><td>0</td><td>0</td></tr><tr><td>Access</td><td>R</td><td>R</td><td>R</td><td>R</td><td>R</td><td>R</td><td>R</td><td>R</td></tr></table>*1 Indicates the status of the external pin IRQ7</td><td>P.13-43</td><td><table><tr><th>Bit No.</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr><tr><td>Bit name</td><td>SC3 TXF</td><td>SC3 RXF</td><td>SC3 TBF</td><td>SC3 RBF</td><td>SC3 CTS</td><td>SC3 FEF</td><td>SC3 PEF</td><td>SC3 OEF</td></tr><tr><td>Reset</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>Access</td><td>R</td><td>R</td><td>R</td><td>R</td><td>R</td><td>R</td><td>R</td><td>R</td></tr></table></td></tr>
<tr><td>P.13-43</td><td></td><td>P.13-43</td><td>(Addition of the following note in the description of Bit No.3.)
Note: When P83A of the port 8 analog/digital input control register P8AD is "1", IRQ7 is treated as "L" internally by the microcontroller and reading the SC3CTS bit returns a value of "0", regardless of the actual values of the port pins.</td></tr>
<tr><td>Chapter 1, Chapter 13</td><td>(Following words in page1-3, 13-3, 13-4, 13-14, 13-15, 13-24, 13-36, 13-45, 13-46, 13-47.)
transfer speed
transfer rate
baud rate</td><td>Chapter 1, Chapter 13</td><td>(In page1-3, 13-3, 13-4, 13-14, 13-15, 13-24, 13-36, 13-45, 13-46, 13-47.)
bit rate</td></tr>
<tr><td>Chapter 1, Chapter 13</td><td>(Following unit in page1-3, 13-3, 13-4, 13-14, 13-15, 13-24, 13-36, 13-46, 13-47.)
bps</td><td>Chapter 1, Chapter 13</td><td>(In page1-3, 13-3, 13-4, 13-14, 13-15, 13-24, 13-36, 13-46, 13-47.)
bit/s</td></tr>
<tr><td>P.14-3</td><td>• Conversion accuracy 10 bits ±4 LSB (Linearity error)</td><td>P.14-3</td><td>• Conversion accuracy 10 bits ±5 LSB (Linearity error)</td></tr>
<tr><td>P.15-47</td><td>[Note 2]
When pin Nos. 48 to 45, respectively, are ...</td><td>P.15-47</td><td>[Note 2]
When pin Nos. 45 to 48, respectively, are ...</td></tr>
<tr><td>P.16-8</td><td>(In the table 16-6-1.) Flash on-board write control register</td><td>P.16-8</td><td>(In the table 16-6-1.) Flash on-board rewrite control register</td></tr>
<tr><td>P.16-8</td><td>(Name of FAREG in the table 16-6-1.) Flash address register</td><td>P.16-8</td><td>(Name of FAREG in the table 16-6-1.) Flash address register (Lower)</td></tr>
<tr><td>P.16-8</td><td>(Name of FAREGEX in the table 16-6-1.) Flash address register</td><td>P.16-8</td><td>(Name of FAREGEX in the table 16-6-1.) Flash address register (Upper)</td></tr>
<tr><td>P.16-8</td><td>(In the table 16-6-1.) Flash on-board write enable register</td><td>P.16-8</td><td>(In the table 16-6-1.) Flash on-board rewrite enable register</td></tr>
<tr><td>P.17-2</td><td>(From the 3rd line of [Ordering method 1].)
... for the user NMI processing routine. This is not necessary if NMIs are not being used, however.)</td><td>P.17-2</td><td>(From the 3rd line of [Ordering method 1].)
... for the user non-maskable interrupt processing routine. This is not necessary if non-maskable interrupts are not being used, however.)</td></tr>
<tr><td>P.17-2</td><td>(The 4th line of [Ordering method 2].)
...when an NMI ...</td><td>P.17-2</td><td>(The 4th line of [Ordering method 2].)
...when a non-maskable interrupt ...</td></tr>
<tr><td>P.17-2</td><td>(The 6th line of [Ordering method 2].)
... of the user NMI processing routine ...</td><td>P.17-2</td><td>(The 6th line of [Ordering method 2].)
... of the user non-maskable interrupt processing routine ...</td></tr>
<tr><td>P.17-2</td><td>(The 7th line of [Ordering method 2].)
If the user is not using NMI processing, ...</td><td>P.17-2</td><td>(The 8th line of [Ordering method 2].)
If the user is not using non-maskable interrupt processing, ...</td></tr>
<tr><td>P.17-3</td><td>(In the fig.17-2-2.) the NMI processing routine</td><td>P.17-3</td><td>(In the fig.17-2-2.) the non-maskable interrupt processing routine</td></tr>
<tr><td>Appendix-10</td><td><table><tr><th></th><th>Source</th><th>Destination</th><th>Format</th><th>Code length</th><th>Execution Cycle</th></tr><tr><td>PUTX</td><td>Dm</td><td></td><td>D0</td><td>2</td><td>2</td></tr><tr><td>PUTCX</td><td></td><td></td><td>D0</td><td>2</td><td>2</td></tr><tr><td>GETX</td><td></td><td>Dn</td><td>D0</td><td>2</td><td>2</td></tr><tr><td>GETCHX</td><td></td><td></td><td>D0</td><td>2</td><td>2</td></tr><tr><td>GETCLX</td><td></td><td></td><td>D0</td><td>2</td><td>2</td></tr><tr><td>MCST</td><td>Dm</td><td>Dn</td><td>D0</td><td>2</td><td>2</td></tr><tr><td>MCST9</td><td></td><td>Dn</td><td>D0</td><td>2</td><td>2</td></tr></table></td><td>Appendix-10</td><td><table><tr><th></th><th>Source</th><th>Destination</th><th>Format</th><th>Code length</th><th>Execution Cycle</th></tr><tr><td>PUTX</td><td>Dm</td><td></td><td>D0</td><td>2</td><td>2</td></tr><tr><td>PUTCX</td><td>Dm</td><td>Dn</td><td>D0</td><td>2</td><td>2</td></tr><tr><td>GETX</td><td></td><td>Dn</td><td>D0</td><td>2</td><td>2</td></tr><tr><td>GETCHX</td><td></td><td>Dn</td><td>D0</td><td>2</td><td>2</td></tr><tr><td>GETCLX</td><td></td><td>Dn</td><td>D0</td><td>2</td><td>2</td></tr><tr><td>MCST</td><td>Dm</td><td>Dn</td><td>D0</td><td>2</td><td>2</td></tr><tr><td>MCST</td><td>imm8</td><td>Dn</td><td>D0</td><td>2</td><td>2</td></tr><tr><td>MCST9</td><td></td><td>Dn</td><td>D0</td><td>2</td><td>2</td></tr></table></td></tr>
<tr><td>–</td><td></td><td>–</td><td>(In addition to these corrections, how to describe the unit is changed , but the data are not changed.

Example) P.12-2, 2nd line of "12.2 Features"
Error : 8 to 15 MHz Correction : 8 MHz to 18 MHz)</td></tr>
</table>

Page	Errors	Page	Corrections
–		–	(All of the series name in this manual is changed as shown below: • "MN10300 Series" is changed the name into "MN1030 Series". • "MN10200 Series" is changed the name into "MN102 Series".)
Warning	The MN1030F01K is manufactured and marketed under a licensing agreement with Bull CP8 Corporation. Note that the MN1030F01K cannot be used on IC cards.	Warning	(Deleted.)
Warning	If you have any inquiries or questions about this book or our semiconductors, please contact one of our sales offices listed at the back of this book or Matsushita Electronics Corporation's Sales Department.	Warning	If you have any inquiries or questions about this book or our semiconductors, please contact one of our sales offices listed at the back of this book.
Colophon	Issued by Matsushita Electric Industrial Co., Ltd. Matsushita Electronics Corporation © Matsushita Electric Industrial Co., Ltd. © Matsushita Electronics Corporation	Colophon	Issued by Matsushita Electric Industrial Co., Ltd. © Matsushita Electric Industrial Co., Ltd.

MN103001G/F01K
LSI User's Manual

February, 2002 5th Edition

Issued by Matsushita Electric Industrial Co., Ltd.

Semiconductor Company, Matsushita Electric Industrial Co., Ltd.

Nagaokakyo, Kyoto 617-8520, Japan
Tel: (075) 951-8151
http://www.panasonic.co.jp/semicon/

SALES OFFICES

■ NORTH AMERICA

●U.S.A. Sales Office:
Panasonic Industrial Company [PIC]

• New Jersey Office:
Two Panasonic Way Secaucus, New Jersey 07094 U.S.A.
Tel: 1-201-348-5257 Fax:1-201-392-4652

• Chicago Office:
1707 N. Randall Road Elgin, Illinois 60123-7847 U.S.A.
Tel: 1-847-468-5720 Fax:1-847-468-5725

• Milpitas Office:
1600 McCandless Drive Milpitas, California 95035 U.S.A.
Tel: 1-408-942-2912 Fax:1-408-946-9063

• Atlanta Office:
1225 Northbrook Parkway Suite 1-151 Suwanee, GA
30024 U.S.A.
Tel: 1-770-338-6953 Fax:1-770-338-6849

• San Diego Office:
9444 Balboa Avenue, Suite 185, San Diego, California
92123 U.S.A.
Tel: 1-619-503-2903 Fax:1-858-715-5545

●Canada Sales Office:
Panasonic Canada Inc. [PCI]
5770 Ambler Drive 27 Mississauga, Ontario, L4W 2T3
CANADA
Tel: 1-905-238-2101 Fax:1-905-238-2414

■ LATIN AMERICA

●Mexico Sales Office:
Panasonic de Mexico, S.A. de C.V. [PANAMEX]
Amores 1120 Col. Del Valle Delegacion Benito Juarez
C.P. 03100 Mexico, D.F. MEXICO
Tel: 52-5-488-1000 Fax:52-5-488-1073

• Guadalajara Office:
SUCURSAL GUADALAJARA
Av. Lazaro Cardenas 2305 Local G-102 Plaza Comercial
Abastos; Col. Las Torres Guadalajara, Jal. 44920
MEXICO
Tel: 52-3-671-1205 Fax:52-3-671-1256

●Brazil Sales Office:
Panasonic do Brasil Ltda. [PANABRAS]
Caixa Postal 1641, Sao Jose dos Campos, Estado de Sao
Paulo
Tel: 55-12-335-9000 Fax:55-12-331-3789

■ EUROPE

●U.K. Sales Office:
Panasonic Industrial Europe Ltd. [PIEL]
Willoughby Road, Bracknell, Berks., RG12 8FP,
THE UNITED KINGDOM
Tel: 44-1344-85-3671 Fax:44-1344-85-3853

●Germany Sales Office:
Panasonic Industrial Europe GmbH [PIEG]
Hans-Pinsel-Strasse 2 85540 Haar, GERMANY
Tel: 49-89-46159-119 Fax:49-89-46159-195

■ ASIA

●Singapore Sales Office:
Panasonic Semiconductor of South Asia [PSSA]
300 Beach Road, #16-01, The Concourse, Singapore
199555 THE REPUBLIC OF SINGAPORE
Tel: 65-390-3688 Fax:65-390-3689

●Malaysia Sales Office:
Panasonic Industrial Company (M) Sdn. Bhd. [PICM]

• Head Office:
Tingkat 16B, Menara PKNS Petaling Jaya, No.17, Jalan
Yong Shook Lin 46050 Petaling Jaya, Selangor Darul
Ehsan, MALAYSIA
Tel: 60-3-7951-6601 Fax:60-3-7954-5968

• Penang Office:
Suite 20-07,20th Floor, MWE Plaza, No.8, Lebuh
Farquhar,10200 Penang, MALAYSIA
Tel: 60-4-201-5113 Fax:60-4-261-9989

• Johore Sales Office:
Menara Pelangi, Suite8.3A, Level8, No.2, Jalan Kuning
Taman Pelangi, 80400 Johor Bahru, Johor, MALAYSIA
Tel: 60-7-331-3822 Fax:60-7-355-3996

●Thailand Sales Office:
Panasonic Industrial (THAILAND) Ltd. [PICT]
252-133 Muang Thai-Phatra Complex Building, 31st Fl.
Rachadaphisek Rd., Huaykwang, Bangkok 10320,
THAILAND
Tel: 66-2-693-3428 Fax:66-2-693-3422

●Philippines Sales Office: [PISP]
Panasonic Indsutrial Sales Philippines Division of Matsushita Electric Philippines Corporation
102 Laguna Boulevard,Bo.Don Jose Laguna Technopark,
Santa. Rosa, Laguna 4026 PHILIPPINES
Tel: 63-2-520-8615 Fax:63-2-520-8629

●India Sales Office:
National Panasonic India Ltd. [NPI]
E Block, 510, International Trade Tower Nehru Place, New
Delhi_110019 INDIA
Tel: 91-11-629-2870 Fax:91-11-629-2877

●Indonesia Sales Office:
P.T.MET & Gobel [M&G]
JL. Dewi Sartika (Cawang 2) Jakarta 13630, INDONESIA
Tel: 62-21-801-5666 Fax:62-21-801-5675

●China Sales Office:
Panasonic Industrial (Shanghai) Co., Ltd. [PI(SH)]
Floor 6, Zhong Bao Mansion, 166 East Road Lujian Zui,
PU Dong New District, Shanghai, 200120 CHINA
Tel: 86-21-5866-6114 Fax:86-21-5866-8000

Panasonic Industrial (Tianjin) Co., Ltd. [PI(TJ)]
Room No.1001, Tianjin International Building 75, Nanjin
Road, Tianjin 300050, CHINA
Tel: 86-22-2313-9771 Fax:86-22-2313-9770

Panasonic SH Industrial Sales (Shenzhen) Co., Ltd. [PSI(SZ)]
7A-107, International Bussiness & Exhibition Centre,
Futian Free Trade Zone, Shenzhen 518048, CHINA
Tel: 86-755-359-8500 Fax:86-755-359-8516

Panasonic Shun Hing Industrial Sales (Hong Kong) Co., Ltd. [PSI(HK)]
11th Floor, Great Eagle Center 23 Harbour Road,
Wanchai, HONG KONG
Tel: 852-2529-7322 Fax:852-2865-3697

●Taiwan Sales Office:
Panasonic Industrial Sales (Taiwan) Co.,Ltd. [PIST]

• Head Office:
6F, 550, Sec. 4, Chung Hsiao E. RD. Taipei, 110, TAIWAN
Tel: 886-2-2757-1900 Fax:886-2-2757-1906

• Kaohsiung Office:
6th Floor, Hsin Kong Bldg. No.251, Chi Hsien 1st Road
Kaohsiung 800, TAIWAN
Tel: 886-7-346-3815 Fax:886-7-236-8362

●Korea Sales Office:
Panasonic Industrial Korea Co., Ltd. [PIKL]
Kukje Center Bldg. 11th Fl., 191 Hangangro 2ga,
Youngsan-ku, Seoul 140-702, KOREA
Tel: 82-2-795-9600 Fax:82-2-795-1542

Printed in JAPAN